AD NUMBER
AD889220
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; 12 AUG 1971. Other requests shall be referred to Air Force Flight Dynamics Laboratory, Attn: FY, Wright-Patterson AFB, OH 45433.**
AUTHORITY
AFFDL ltr, 29 Nov 1973

AFFDL-TR-71-7
Volume II

Vol I - AD 889039L

A STUDY OF FOLDING PROPROTOR VTOL AIRCRAFT DYNAMICS

VOLUME II. COMPUTER PROGRAMS

AD 889220

H. E. LOSEY
P. Y. HSIEH

BELL HELICOPTER COMPANY

TECHNICAL REPORT AFFDL-TR-71-7, VOLUME II

SEPTEMBER 1971

Distribution limited to U.S. Government agencies only; test and evaluation; statement applied 12 August 1971. Other requests for this document must be referred to AF Flight Dynamics Laboratory, (FY), Wright-Patterson Air Force Base, Ohio 45433.

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

254

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE: 27-10-71/100

A STUDY OF FOLDING PROPROTOR VTOL AIRCRAFT DYNAMICS

VOLUME II. COMPUTER PROGRAMS

H. E. LOSEY
P. Y. HSIEH
BELL HELICOPTER COMPANY

FOREWORD

This report was prepared by Bell Helicopter Company of Fort Worth, Texas for the Aerospace Dynamics Branch, Vehicle Dynamics Division, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio under Contract F33615-69-C-1339. This research is part of a continuing dynamic and aeroelastic phenomena for rotor/propeller powered V/STOL flight vehicles under the Air Force Systems Command's exploratory development program. The Project Number is 1370, "Dynamic Problems in Military Flight Vehicles", Task Number is 137005, "Prediction and Control of Flight Vehicle Vibration". Mr. A. R. Basso of the Aerospace Dynamics Branch was the Project Engineer.

The final report is presented in two volumes. The first volume describes the development of analytical methods of predicting folding proprotor (FPR) aircraft dynamic behavior, correlation of these methods with model test data, and the results of the dynamic analysis of a representative FPR design. The second volume is a guide to the digital computer programs and contains input and output formats and FORTRAN listings of the programs.

Mr. Troy Gaffey was the Bell Helicopter Company Project Engineer. The authors acknowledge the assistance of Dr. Jing Yen and Mr. Gottfried Weber in providing the equations of motion and in checking out the programs.

This report covers work conducted from February 1969 to February 1971. Manuscript was released by authors in February 1971 for publication as an AFFDL Technical Report.

This technical report has been reviewed and is approved.

Walter J. Mykytow

WALTER J. MYKYTOW
Asst. for Research & Technology
Vehicle Dynamics Division

ABSTRACT

This report describes the results of a study of the dynamic stability and response during the feathering and folding of the blades of a folding-proprotor VTOL. This study involved the development of analytical methods of predicting the dynamic characteristics during feathering and folding, correlation of the theoretical methods with experimental data, a dynamic analysis of a representative folding-proprotor VTOL design, and a parametric study to identify design factors which can be used to control dynamic characteristics during feathering and folding. Correlation of the theory with measured dynamic stability and response characteristics during feathering and folding is good, and indicates the analytical methods can be used with confidence. The results of the dynamic analysis indicate satisfactory stability and response characteristics can be achieved in a 66,000-pound gross-weight class, folding-proprotor VTOL. The results of the parametric study are summarized in terms of design guidelines. This volume is a guide to the computer programs, and contains FORTRAN listings. Volume I contains the development of the theory, correlation of theory with experimental data, and the dynamic response and parametric study.

TABLE OF CONTENTS

VOLUME II

LIST OF ILLUSTRATIONS

VOLUME II

LIST OF TABLES

Volume II

LIST OF SYMBOLS

AC	aerodynamic center
a,b,c	coefficients
a_i	estimated lift curve slope
B	reference semichord
CG	center of gravity
C_T	chord tip
$D\overline{Y}_i$	actual geometric strip width
EA	elastic axis
EI	segment beam or chord stiffness
e	chord root divided by chord tip
F(t)	forcing function
$\{F\}$	force matrix on a wing
GJ	segment torsional stiffness
h	integration interval
1/K	reduced velocity
$\lceil k \rfloor$	modified wing stiffness matrix
S	Vt divided by B
SIC	flexibility matrix
t	time
V	forward velocity
x	variable
$\{x\}$	wing normal coordinate matrix
x_c	calculated value of x
x_g	guessed value of x

α	local angle of attack
ε	error tolerance
$\partial L/\partial \alpha$	lift curve slope
Δx	$x_c - x_g$
λ_R	real part of the eigenvalue
Θ	element pitch

SECTION I

INTRODUCTION

This volume provides a guide to the use of the computer programs developed during this study. It also contains the information necessary to modify the programs or to adapt them to a given computer. A FORTRAN listing of the programs is included in this volume. It should be noted that the program listings are the only source for the complete equations of motion. Section II of Volume I provides an overview of the theory mechanized in the programs.

Three computer programs were developed: ARAP06, which treats that portion of the FPR transition mode where the blades are rotating, and DFAL17 and DFAL18, which treat that portion where the proprotor has been stopped and the blades are being folded. ARAP06 is all inclusive in that it may be used to calculate the stability, loads, and response in proprotor mode and during blade feathering. DFAL17 is used to calculate the system stability during folding; DFAL18, the loads and response during folding.

Sample cases for each program are contained in Appendix I. These include card data and condensed output.

SECTION II

USER GUIDE TO PROGRAMS

In preparing this section it is assumed that the reader has read Section II of Volume I and is familiar with the math model and capabilities and limitations of each program.

This user's guide consists of four parts for each program: (1) an input format identifying the information contained on each card, (2) an input format guide explaining the function and source of each input, (3) an output format and guide, and (4) suggested techniques for using the programs.

A. ARAP06 - FPR FEATHERING DYNAMICS ANALYSIS

The inputs to ARAP06 consist of two types of data: (1) math model configuration and control options and (2) vehicle descriptive data. The options allow the user to select the math model, the type of excitation and/or the maneuver to be considered, and the output data. The descriptive data consist of the geometry, certain spring rates, mass distributions, and the coupled normal modes of the subject vehicle.

To generate the normal mode input data the user must have access to a coupled beam-chord-torsion vibration analysis (to determine the normal modes of the wing/pylon system) and to a rotating beam vibration analysis (to determine the blade normal modes).

Because of the various options with regard to the math model, all of the input data will generally not be required. For example, in many cases the equations representing blade elastic motion may be deleted without significantly influencing the results. In those cases blade normal mode data are not required as a program input. Careful selection of the options for a given case can greatly reduce the data preparation effort.

Figure 1 illustrates the card order in a ARAP06 data deck.

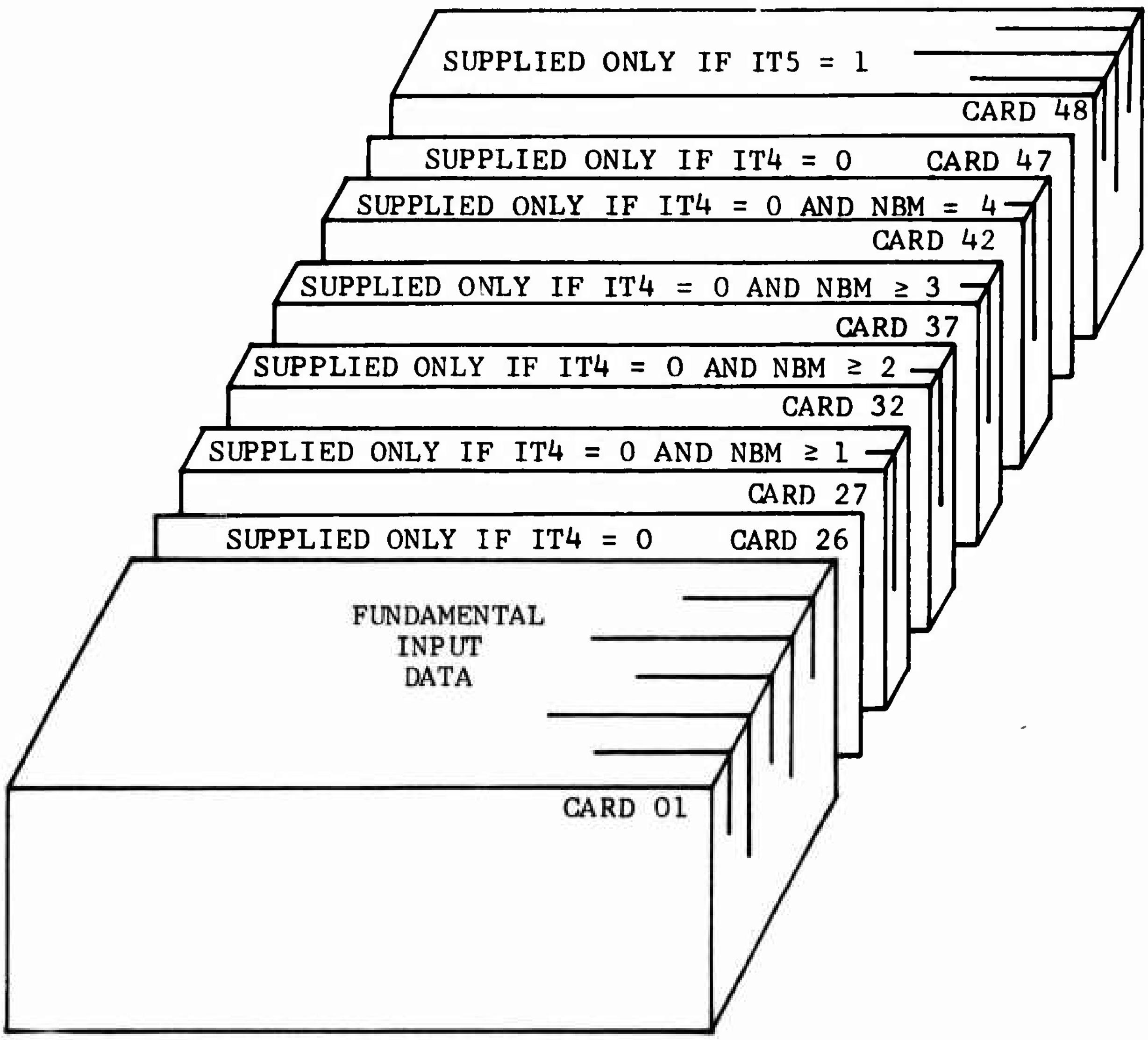


Figure 1. Card Order in ARAP06 Data Deck.

1. Input Format

CARD 01 Title Card; Column 1-80, alphameric.

CARD 02 Format 16I5:

Col. 1 - 5	IT1
Col. 6 - 10	IT2
Col. 11 - 15	IT3
Col. 16 - 20	IT4
Col. 21 - 25	IT5
Col. 26 - 30	IT6
Col. 31 - 35	IT7
Col. 36 - 40	IT8
Col. 41 - 45	IT9
Col. 46 - 50	IT10
Col. 51 - 55	IT11
Col. 56 - 60	IT12
Col. 61 - 65	IT13
Col. 66 - 70	IT14
Col. 71 - 75	IT15

CARD 03 Format 16I5:

Col. 1 - 5	NS
Col. 6 - 10	NBM
Col. 11 - 15	NMT
Col. 16 - 20	NTHC
Col. 21 - 25	NPSID
Col. 26 - 30	NCASE

CARD 04 Format 8F10.0:

(1)	Pylon length	(in)
(2)	Speed of sound	(ft/sec)
(3)	Air density	(slug/ft^3)
(4)	Pylon conversion angle	(deg)
(5)	Structural damping ratio of rotor-blades	
(6)	Rotor rpm	(rpm)
(7)	Blade fundamental torsional natural frequency	(rad/sec)
(8)	Gravitational acceleration	(in/sec^2)

CARD 05 Format 8F10.0:

(1)	Blade built-in precone angle	(deg)
(2)	Blade pitch-flap coupling, (δ_3)	
(3)	Blade collective pitch at initial rpm	(deg)
(4)	Blade lateral cyclic pitch	(deg)
(5)	Blade longitudinal cyclic pitch	(deg)
(6)	Aircraft true airspeed	(kt)
(7)	Real time limit	(sec)
(8)	Gravitational acceleration	(in/sec^2)

CARD 06 Format 8F10.0:

(1)	Integration interval	(sec)
(2)	Initial azimuth position of master blade	(deg)
(3)		
(4)		
(5)		
(6)		
(7)		
(8)		

CARD 07 Format 8F10.0:

(1)	RPM 1	(rpm)
(2)	RPM 2	(rpm)
(3)	RPM 3	(rpm)
(4)	RPM 4	(rpm)
(5)	Longitudinal hub restraint spring rate for rpm 1	(ft-lb/deg)
(6)	Longitudinal hub restraint spring rate for rpm 2	(ft-lb/deg)
(7)	Longitudinal hub restraint spring rate for rpm 3	(ft-lb/deg)
(8)	Longitudinal hub restraint spring rate for rpm 4	(ft-lb/deg)

CARD 08 Format 8F10.0:

(1)	Lateral hub restraint spring rate for rpm 1	(ft-lb/deg)
(2)	Lateral hub restraint spring rate for rpm 2	(ft-lb/deg)
(3)	Lateral hub restraint spring rate for rpm 3	(ft-lb/deg)
(4)	Lateral hub restraint spring rate for rpm 4	(ft-lb/deg)
(5)	Flapping stop spring rate	(ft-lb/deg)
(6)	Blade flapping amplitude at which flapping stop is encountered	(deg)
(7)		
(8)		

CARD 09 Format 8F10.0:

(1)	Blade collective pitch at which feathering or unfeathering is to end	(deg)
(2)	Real time at which feathering or unfeathering starts	(sec)
(3)	Real time at which feathering or unfeathering ends	(sec)
(4)		

(5)
(6)
(7)
(8)

CARD 10	Format 8F10.0:		
	(1)	Amplitude of vertical gust	(in/sec)
	(2)	Amplitude of head-on gust	(in/sec)
	(3)	Frequency of sinusoidal gust	(cps)
	(4)	Real time at which sharp-edged gust starts	(sec)
	(5)	Real time at which sharp-edged gust ends	(sec)
	(6)	Real time at which 1-cosine shaped gust starts	(sec)
	(7)	Real time at which 1-cosine shaped gust ends	(sec)
	(8)		
CARD 11	Format 8F10.0:		
	(1)	Wing angle of attack	(deg)
	(2)	Mean aerodynamic chord of wing	(in)
	(3)	Two-dimensional lift curve slope of wing	(/rad)
	(4)	Wing thickness ratio	
	(5)	Wing sweep angle	(deg)
	(6)	Distance between pylon conversion hinge and leading edge of wing at wing tip	(in)
	(7)	Distance between shaft hinge and wing tip section	(in)
	(8)		
CARD 12	Format 8F10.0:		
	(1)	Initial frequency of cosine function for swashplate lateral excitation	(cps)
	(2)	Initial frequency of sine function for swashplate lateral excitation	(cps)
	(3)	Initial frequency of cosine function for swashplate longitudinal excitation	(cps)
	(4)	Initial frequency of sine function for swashplate longitudinal excitation	(cps)
	(5)	Amplitude of cosine function for function for swashplate lateral excitation	(deg)
	(6)	Amplitude of sine function for swash-plate lateral excitation	(deg)
	(7)	Amplitude of cosine function for swashplate longitudinal excitation	(deg)
	(8)	Amplitude of sine function for swash-plate longitudinal excitation	(deg)

CARD 13 Format 8F10.0:

(1)	Rate-of-change of frequency of cosine function for swashplate lateral excitation	(cyc/sec^2)
(2)	Rate-of-change of frequency of sine function for swashplate lateral excitation	(cyc/sec^2)
(3)	Rate-of-change of frequency of cosine function for swashplate longitudinal excitation	(cyc/sec^2)
(4)	Rate-of-change of frequency of sine function for swashplate longitudinal excitation	(cyc/sec^2)
(5)	Real time at which both excitations of swashplate start	(sec)
(6)		
(7)		
(8)		

CARD 14 Format 8F10.0:

(1)	Component in a cartesian coordinate to define the location of an aerodynamic vane	(in)
(2)	Component in a cartesian coordinate to define the location of an aerodynamic vane	(in)
(3)	Component in a cartesian coordinate to define the location of an aerodynamic vane	(in)
(4)	Span of aerodynamic vane	(in)
(5)	Chord of aerodynamic vane	(in)
(6)	Initial frequency for aerodynamic vane excitation	(cps)
(7)	Steady angle-of-attack of aerodynamic vane, relative to pylon	(deg)
(8)	Amplitude of angle-of-attack oscillation of aerodynamic vane, relative to pylon	(deg)

CARD 15 Format 8F10.0:

(1)	Real time at which excitation of aerodynamic vane starts	(sec)
(2)	Two-dimensional lift curve slope for aerodynamic vane	(/deg)
(3)	Rate-of-change of the excitation frequency for aerodynamic vane	(cyc/sec^2)
(4)		
(5)		
(6)		
(7)		
(8)		

CARD 16*	Format 8F10.0: Blade build-in twists	(deg)
CARD 17*	Format 8F10.0: Radial distances to blade discrete masses	(in)
CARD 18*	Format 8F10.0: Discrete blade masses	(chug)**
CARD 19*	Format 8F10.0: Blade mass unbalances	(in)
CARD 20*	Format 8F10.0: Blade torsional mass moments-of-inertia	(chug-in^2)
CARD 21*	Format 8F10.0: Distances from blade feathering axis to blade aerodynamic center	(in)
CARD 22*	Format 8F10.0: Distances from blade aerodynamic center to blade center of gravity	(in)
CARD 23*	Format 8F10.0: Distances from blade aerodynamic center to blade elastic axis	(in)
CARD 24*	Format 8F10.0: Blade chord lengths	(in)
CARD 25*	Format 8F10.0: Blade aerodynamic segment lengths	(in)
CARD 26	Format 13F6.0, 2X: Blade root collective pitches followed by rotor angular velocities	(deg & rad/sec)
CARD 27	Col. 1 - 80, alphameric. Title card to identify blade first normal mode	
CARD 28*	Format 13F6.0, 2X: Blade natural frequencies for first normal mode	(rad/sec)

* Indicates more than one card may be required for these data.
**A unit of mass. The corresponding unit of acceleration is in inches per second^2.

CARD 29	Format 16I5: Col. 1 - 5 LC(1) Col. 6 - 10 LC(2) Col. 11 - 15 LC(3) Col. 16 - 20 KDR	
CARD 30*	Format 13F6.0, 2X: Normalized blade flexible out-of-plane, in-plane, and torsional modes	(in/in, in/in, rad/in)
CARD 31*	Format 9F8.0, 8X: Blade normalized beamwise bending moments and chordwise bending moments	(in-lb/in)
CARD 32 CARD 36	Same as cards 27-31, except now for the second normal mode	
CARD 37 CARD 41	Same as cards 27-31, except now for the third normal mode	
CARD 42 CARD 46	Same as cards 27-31, except now for the fourth normal mode	
CARD 47	Format 13F6.0, 2X: Initial conditions of normal coordinates for flexible blades	
CARD 48	Format 16I5: Col. 1 - 5 NSW Col. 6 - 10 NSHP Col. 11 - 15 NPDD	
CARD 49	Format 8F10.0: (1) Wing structural damping ratio	
	(2) Pylon mass	(chug)
	(3) Component of distance from pylon conversion axis to pylon center of gravity in cartesian coordinates	(in)
	(4) Component of distance from pylon conversion axis to pylon center of gravity in cartesian coordinates	(in)
	(5) Component of distance from pylon conversion axis to pylon center of gravity in cartesian coordinates	(in)

*Indicates more than one card may be required for these data

	(6) Hub mass	(chug)
	(7) Mean distance from wing aerodynamic center to its elastic axis	(in)
	(8) Pylon pitching mass moment-of-inertia about its center of gravity	(chug-in^2)
CARD 50	Format 8F10.0: Initial conditions for wing normal coordinates	
CARD 51*	Format 8F10.0: Wing chordwise normal modes	(in/in)
CARD 52*	Format 8F10.0: Wing beamwise normal modes	(in/in)
CARD 53*	Format 8F10.0: Wing torsional normal modes	(rad/in)
CARD 54*	Format 8F10.0: Discrete wing masses, wing torsional mass moments-of-inertia, wing mass unbalances, and wing aerodynamic segment lengths	 (chug, chug-in^2, in, in)
CARD 55*	Format 8F10.0: Wing/pylon natural frequencies; normalized F/A deflections, lateral deflections, and vertical deflections at CA Normalized pylon yawings, pitchings, and effective masses	 (rad/sec, in/in, in/in, in/in, rad/in, rad/in, chug)
CARD 56*	Format 5D15.0: Elements of stiffness matrix at a specified wing buttline	
CARD 57*	Format 8F10.0: Locations of pylon accelerometers, relative to pylon conversion axis	(in)

*Indicates more than one card may be required for these data

2. Guide to Input Format

CARD 01 This card is used to identify the output.

CARD 02 This card is used to specify the options desired. The options selected will depend on the configuration simulated and on the users objectives. It is pointed out that much of the other data which must be supplied is dependent to some extent on the options selected. Any discrepancy between the data requirements of the options selected and the data supplied will cause the program to terminate.

IT1 allows the hub restraint springs used with a gimbaled rotor to be simulated.

If IT1 = 0, hub restraint springs are not simulated.

If IT1 = 1, hub restraint springs are simulated.

IT2 is used to feather or unfeather the blades to start or stop the proprotor.

If IT2 = 0, the blades will be feathered or unfeathered.

If IT2 = 1, the blade collective pitch will be constant.

IT3 is used to simulate a discrete gust encounter or an oscillation of a wind tunnel airstream such as in the NASA-Langley 16-foot TDT.

If IT3 = 0, no gust will be encountered.

If IT3 = 1, a sharp edged gust will be encountered.

If IT3 = 2, a 1-cosine gust of specified length will be encountered.

If IT3 = 3, a continuous, sinusoidal gust is applied.

IT4 allows the program to be run with or without the blade elastic degrees of freedom.

If IT4 = 0, the blade elastic degrees of freedom will be simulated (cards 26-47 must be supplied).

If IT4 = 1, the blades are rigid (cards 26-47 must be deleted).

IT5 allows the program to be run with or without the wing/pylon elastic degrees of freedom.

If IT5 = 0, the wing degrees of freedom are deleted (cards 48 through end of deck must be deleted).

If IT5 = 1, the wing/pylon degrees of freedom are simulated (cards 48 through end of deck must be supplied).

IT6 gives the user a choice as to machine plotted data.

If IT6 = 0, plots will not be generated.

If IT6 = 1, time histories will be plotted for rotor rpm, longitudinal and lateral flapping, rotor H-force, rotor Y-force, rotor thrust, and torque.

If IT6 = 2, time histories will be plotted for the inplane and out-of-plane deflections of the master blade, beamwise and chordwise bending moments at the first specified blade station, beamwise and chordwise moments at the second specified blade station and flapping in the rotating system.

If IT6 = 3, time histories will be plotted for the wing tip beamwise, chordwise and, torsional deflections, vertical accelerations of the first two specified accelerometers, and the axial and lateral accelerations at the first accelerometer.

IT7 allows the aerodynamic interference between the wing and rotor to be simulated.

If IT7 = 0, the aerodynamic interference will be zero.

If IT7 = 1, the aerodynamic interference will be simulated.

IT8 allows the user to excite the system using either the swashplate or an aerodynamic vane located on the pylon.

If IT8 = 0, there is no excitation.

If IT8 = 1, the swashplate will be excited cyclically.

If IT8 = 2, the aerodynamic vane will be oscillated in pitch.

IT9 allows the user to collect peak values for several variables throughout the time history. This option prints out at the end of the output data the maximum and minimum values of master blade flapping (rotating system), beamwise and chordwise deflections of the wing tip, and the first normal coordinate of the wing/pylon system.

If IT9 = 0, peaks will not be printed out.

If IT9 = 1, peaks are printed out.

IT10 allows the accelerations at specified pylon stations to be calculated and printed out.

If IT10 = 0, the accelerations will not be printed.

If IT10 = 1, the accelerations at up to nine specified locations will be printed. If IT5 = 0, this option is ignored by the program.

IT11 allows wing bending moments to be calculated.

If IT11 = 0, wing bending moments will not be calculated.

If IT11 = 1, wing bending moments at a specified station will be calculated and printed out.

IT12 allows the user a choice of rigid-body freedoms of the proprotor.

If IT12 = 0, a gimbaled semirigid rotor with rotational freedom (variable rpm) is simulated.

If IT12 = 1, the blade rigid-body flapping is locked out to allow simulating of hingeless or articulated rotors. Rotor rpm is a variable.

If IT12 = 2, the rotor rpm is constant (flapping is free).

If IT12 = 3, the rigid-body flapping and rotational freedoms are locked out.

IT13 allows the effective mass of the wing/pylon modes to be calculated by the program.

If IT13 = 0, the wing/pylon normal modes effective masses are calculated internally.

If IT13 = 1, input values of the wing/pylon effective masses are used (card 55).

IT14 is used to overcome a numerical problem often encountered with time history analyses. In some cases pylon accelerations with frequencies the same as one-half the reciprocal of the integration interval have been encountered. By averaging the accelerations to be calculated these high frequency accelerations can be eliminated.

If IT14 = 0, no averaging takes place.

If IT14 = 1, calculated accelerations are averaged.

IT15 is not used at present.

CARD 03 is used to read in parameters which specify the amount of the input data to follow.

NS is the number of blade lumped mass stations. $0 < NS \leq 10$

NBM is the number of normal blade modes used. $0 < NBM \leq 4$. NBM is used only if IT4 = 0.

NMT is the number of blade stations at which blade bending moments are to be calculated. $0 < NMT \leq 4$. NMT is used only if IT4 = 0.

NTHC is the number of blade collective pitch angles used in the blade normal mode tables. $0 < NTHC \leq 6$

	NPSID	is the number of rotor rpm used in the blade normal mode tables. $0 < \text{NPSID} \leq 6$
	NCASE	number of cases in a single computer run. It is read in from the data deck of the first case only. If NCASE is left blank, the program will assume there is only one case.
CARD 04	contains descriptive data.	
	ZH	is the pylon length; the distance from the pylon conversion axis to the rotor hub.
	S	is the speed of sound; must be greater than zero.
	RHO	is the air density. If it is input as zero, computation of rotor aerodynamics, wing aerodynamics, pylon aerodynamic excitation, and the wing/rotor aerodynamic interference is not made. If user does not wish to include rotor aerodynamic forces, zeros should be input on card 25. If user does not wish to include wing aerodynamics, zeros should be input for the wing aerodynamic segment lengths on card 54.
	TPC	is the pylon conversion angle, measured relative to flight path. It is zero for a proprotor in propeller mode, 90 degrees for helicopter mode.
	BZETA	is the blade structural damping ratio, ignored by the program if IT4 = 1.
	RPM	is initial rotor rpm. $\text{rpm} \geq 0$
	OMT	is blade fundamental torsional natural frequency. $\text{OMT} > 0$. OMT is used in the calculation of blade pitch damping coefficients.
	G	is the gravitational acceleration normal to the wing. $G \geq 0$
CARD 05	a_0	is the rotor precone angle.
	δ_3	is the blade pitch-flap coupling. Positive if blade pitch decreases when the blade flaps upward.

θ_c is the blade initial collective pitch. Corresponds to initial rotor rpm.

A_1 is the blade lateral cyclic pitch (sign convention is that given in Reference 1).

B_1 is the blade longitudinal cyclic pitch.

U is the velocity along flight path. Must be greater than or equal to zero.

REALTM the real time required for the case. Program execution is contained in the time history loop until REALTM is reached. To avoid waste of machine time due to input errors the program is terminated if the rotor rotational speed is below -2 rad/sec.

GY Gravitational acceleration parallel to the wing span. For example, GY is used for a wind tunnel floor-mounted semispan model wing.

CARD 06 DTMIN is the time increment used in the time history calculation. The user should not attempt to use a DTMIN which is larger than the equivalent of $\Delta\psi$ = 15 degrees. A value of DTMIN = 0.005 second is picked by the program if it is not specified. Experience suggests the use DTMIN = 0.005 second for runs where the blade normal modes are included. Otherwise the following DTMINs may be used:

For rotor rpm $\geq$ 300	DTMIN = 0.006 second
300 rotor rpm $\geq$ 200	DTMIN = 0.007 second
200 rotor rpm $\geq$ 100	DTMIN = 0.01 second
rotor rpm < 100	DTMIN = 0.012 second

Since averaging process is taken if IT14 = 1, integration interval will then be twice of DTMIN. This doubled value should not be larger than the equivalent of $\Delta\psi$ = 15 degrees.

PSIE is the initial azimuth position of the master blade.

CARD 07 The data on this card can be used to simulate a nonisotropic, variable rate, hub restraint spring. The rpm range is divided into three parts, and the variation of spring rate is assumed to be linear over these three regions, see Figure 2.

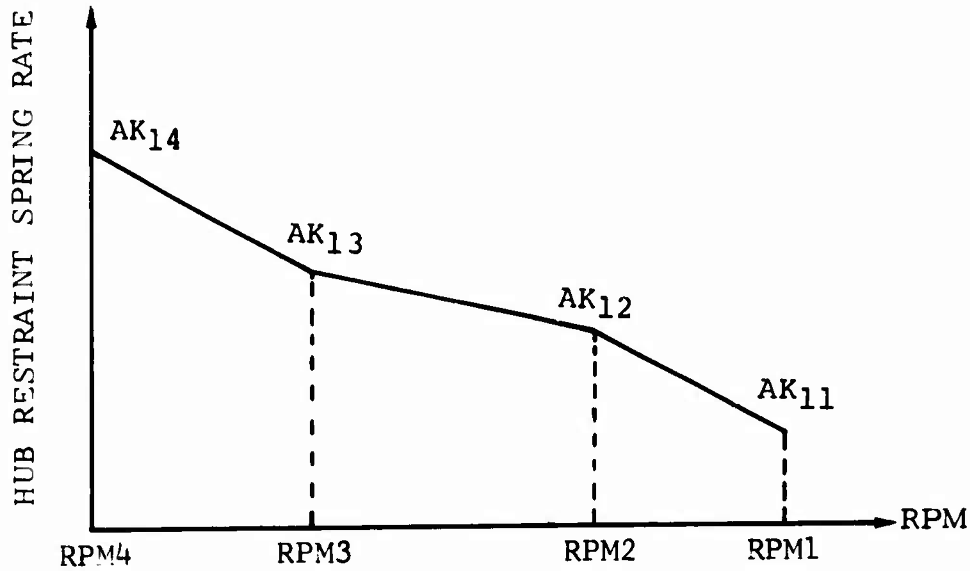


Figure 2. Hub Restraint Spring Rate-rpm Relationship.

The first four floating point numbers, RPM1, RPM2, RPM3, and RPM4 give the rpm at which the spring rate is specified. The last four, AK_{11}, AK_{12}, AK_{13}, and AK_{14}, specify the longitudinal hub spring rate at these rpm.

CARD 08 This card is used in conjunction with Card 07. The first four floating point numbers BK11, BK12, BK13, and BK14 give the lateral hub spring rates at the four rpm specified on Card 07. The fifth, STOPK, is a hub spring rate simulating flapping stops. The sixth, STOPTH, is the blade flapping angle at which the blade (or hub) contacts the flapping stop. If the longitudinal hub spring rate on Card 07 is constant, and the lateral hub spring rate on Card 08 is also constant; it is then suggested that zeros be input for each rpm to avoid an interpolation requirement.

CARD 09 This card controls blade feathering. It is ignored if IT2 = 1.

THCMAX	This is the blade collective pitch at which feathering or unfeathering stops.
TMST	Real time at which feathering or unfeathering starts.
TMED	Real time at which feathering or unfeathering ends.

CARD 10 This card specifies the magnitude and time a gust is encountered. The magnitude and time relationships are illustrated in Figure 3.

(a) Step Gust

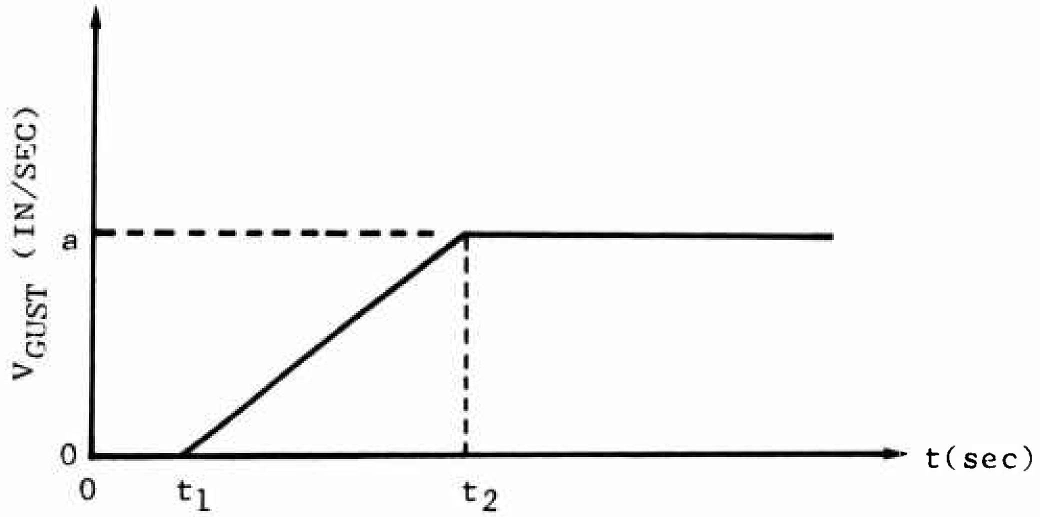
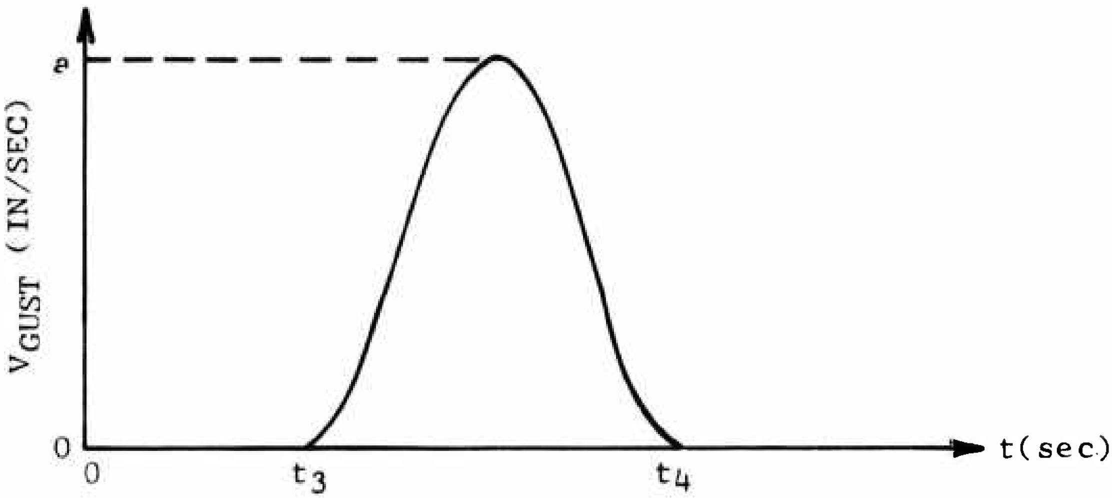


Figure 3. Gust Magnitude and Time Relationships.

(b) 1 - cosine gust

$$V_{gust} = a \sin^2\left[\frac{\pi}{2}\,\frac{t - t_3}{(t_4 - t_3)/2}\right]$$



(c) Sinusoidal gust

$$V_{gust} = a \sin \omega t$$

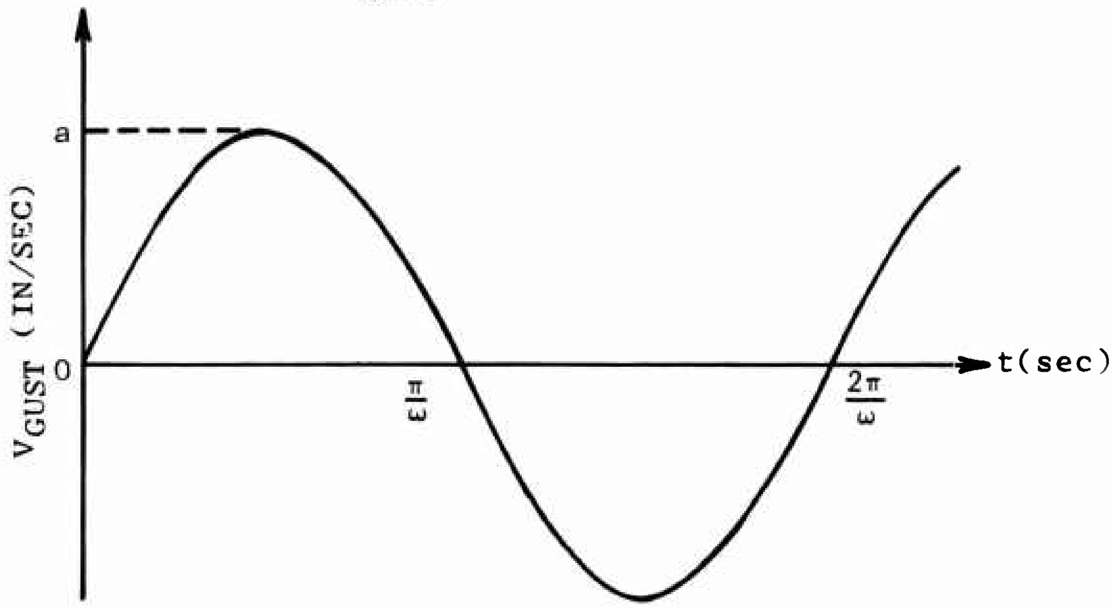


Figure 3. Concluded.

CARD 11 This card is used to specify wing aerodynamic and wing/rotor aerodynamic interference parameters. If IT5 = 0 and IT7 = 0, the data on this card will not be used.

ALW is the wing angle of attack. Positive wing leading-edge up.

WC is the wing mean aerodynamic chord, must be greater than zero.

SLC is the two-dimensional lift curve slope for wing. Stall is not considered.

DEL is the ratio of wing thickness to wing mean chord.

ALAM is the wing sweep angle, positive forward.

AL Illustrated in Figure 3.

YCA Illustrated in Figure 3.

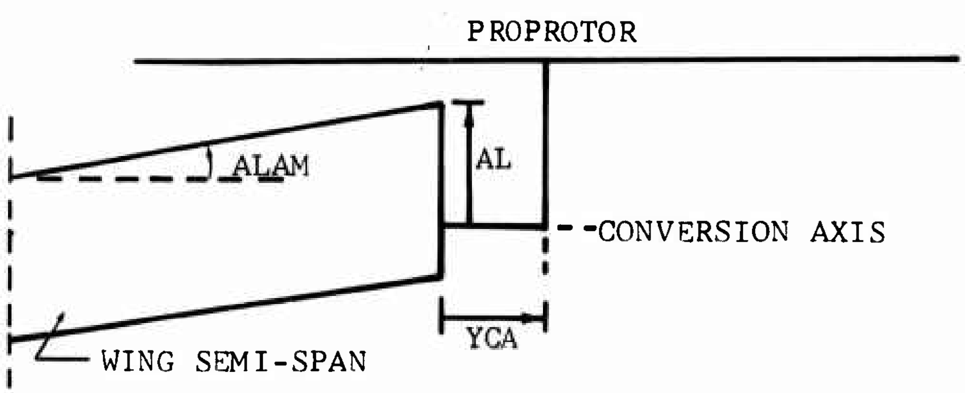


Figure 4. Rotor/Wing Geometrical Relations.

CARD 12 This card and Card 13 are used for swashplate cyclic excitation. For IT8 = 1, the position of the swashplate is given by:

for $t < t_\tau$

$$A_1(t) = A_1$$

$$B_1(t) = B_1$$

for $t \geq t_\tau$

$$A_1(t) = A_1 + A_{1c} \cos\left[\omega_{ac}\left(t-t_\tau\right)\right] + A_{1s} \sin\left[\omega_{as}\left(t-t_\tau\right)\right]$$

$$B_1(t) = B_1 + B_{1c} \cos\left[\omega_{bc}\left(t-t_\tau\right)\right] + A_{1s} \sin\left[\omega_{bs}\left(t-t_\tau\right)\right]$$

where t_τ is the time excitation initiated.

And,

$$\omega_{ac} = \omega_{ac0} + \frac{\partial\omega_{ac}}{\partial t}\left(t-t_\tau\right)$$

$$\omega_{as} = \omega_{as0} + \frac{\partial\omega_{as}}{\partial t}\left(t-t_\tau\right)$$

$$\omega_{bc} = \omega_{bc0} + \frac{\partial\omega_{bc}}{\partial t}\left(t-t_\tau\right)$$

$$\omega_{bs} = \omega_{bs0} + \frac{\partial\omega_{bs}}{\partial t}\left(t-t_\tau\right)$$

The eight floating point numbers on this card are input in the order that: ω_{ac0}, ω_{as0}, ω_{bc0}, ω_{bs0}, A_{1c}, A_{1s}, B_{1c}, and B_{1s}.

CARD 13 Inputs are in the following order: $\partial\omega_{ac}/\partial t$, $\partial\omega_{as}/\partial t$, $\partial\omega_{bc}/\partial t$, $\partial\omega_{bs}/\partial t$, and t_τ.

CARD 14 The data on this card and Card 15 are used for an aerodynamic vane mounted on the pylon to excite the wing/pylon normal modes. The location and dimensions of the exciter can be specified. The angle of attack of the vane is calculated by

$$\alpha_e = \alpha_{pc} + \alpha_{0e} + \alpha_{0s} \sin\left[\omega_e(t-t_e)\right]$$

in which α_{pc} is the pylon conversion angle, α_{0e} and α_{0s} are the steady and oscillatory angle of attack of the vane relative to the pylon respectively, t_e is the time when excitation of the vane begins, and ω_e is the exciting frequency given by:

$$\omega_e = \omega_{e0} + \frac{\partial \omega_e}{\partial t}(t-t_e)$$

UE, VE, WE — are the Cartesian coordinates used to locate the aerodynamic vane in the pylon coordinate system. The intersection of the quarter chord and midspan of the vane is used as the reference point.

SPNE — is the span of the vane.

CDE — is the aerodynamic chord of the vane.

OME — is the ω_{e0} in the expression for ω_e.

ALX — is the α_{0e} in the equation for α_e. Positive if the leading edge of the vane is up.

ALS — is the α_{0s} in the equation for α_e. It is positive if leading edge of the vane is up.

CARD 15 — This card is a continuation of Card 14.

TEX — is the time excitation begins (t_e).

CLXO — is the lift curve slope of the vane.

OMES — is the rate of change of vane frequency, $\partial\omega_e/\partial t$.

CARD 16* — Card 16 through 25 contain blade geometry and inertial data.

THT(I) — is the blade built-in twist for a given blade segment. THT is read in from blade root to blade tip. Washout is negative thrust (I = 1 to NS).

CARD 17* — R(I) — is the radial distance for the blade lumped masses. R(I) is read in from root to tip (I = 1 to NS).

*Indicates more than one card may be required for these data.

CARD 18*	AM(I)	is the blade lumped masses. AM(I) is read in from root to tip (I = 1 to NS).
CARD 19*	YB(I)	is the blade mass unbalance. YB(I) is read in from root to tip (I = 1 to NS).
CARD 20*	AIC(I)	is the blade torsional mass moment of inertia. AIC(I) is read in from root to tip (I = 1 to NS).
CARD 21*	EPF(I)	is the distance from the blade feathering axis to the blade aerodynamic center. Positive if AC is ahead of the feathering axis. Input is read in from root to blade tip (I = 1 to NS).
CARD 22*	EPC(I)	is the distance from the blade aerodynamic center to the blade center of gravity. Positive if the center of gravity is ahead of the aerodynamic center (I = 1 to NS). Read in from blade root to blade tip.
CARD 23*	EPE(I)	is the distance from the blade aerodynamic center to the blade elastic axis. Positive if aerodynamic center is ahead of the elastic axis (I = 1 to NS). Read in from blade root to blade tip.
CARD 24*	C(I)	is the blade segment chord length. Input from root to tip (I = 1 to NS).
CARD 25*	DR(I)	is the width of the blade segments. A zero value of DR(I) can be used to eliminate the blade aerodynamics. By varying the segment lengths at the blade root and tip, blade root cut-out and tip loss may be simulated.

CARDS 26-47 are used to input the blade normal modes. The information on Card 26 is used for all of the modes. Cards 27 through 31 contain data describing the first blade mode. Cards 32 through 36, 37 through 41, and 42 through 46 contain data for the second, third and fourth modes respectively. If IT4 = 1, cards 26 through 47 should be deleted from the data deck. If IT4 = 0, then NBM on Card 03 will determine how many data cards must be deleted from the data deck.

*Indicates more than one card may be required for these data.

The blade normal mode data are supplied in a three-dimensional array format as illustrated in Figure 5.

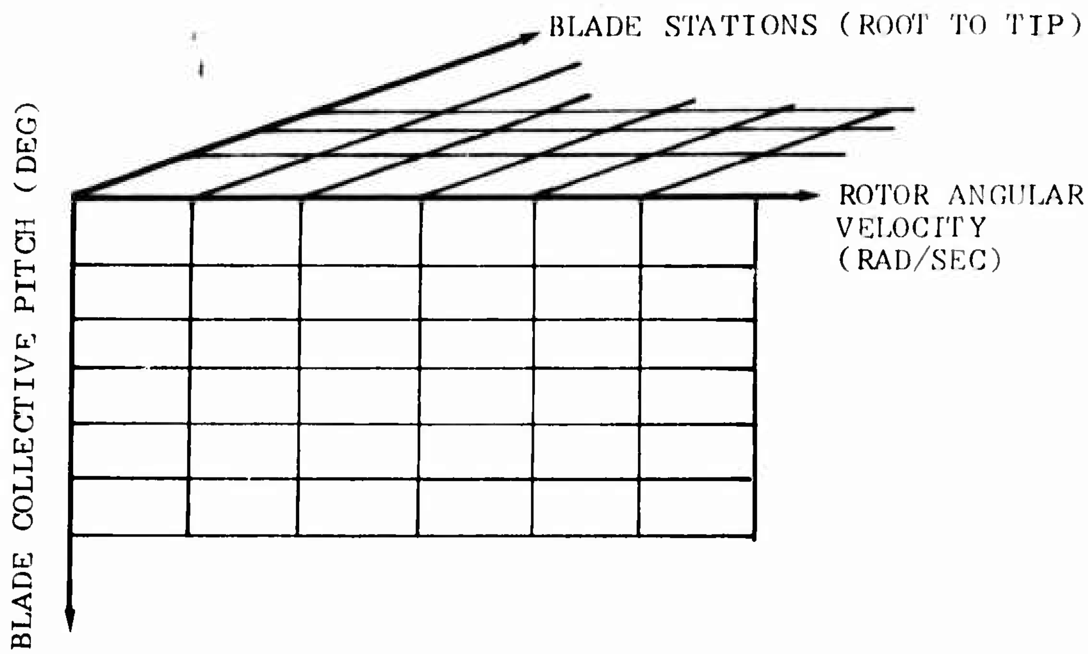


Figure 5. Three-Dimensional Array Format for Blade Normal Modes Data.

CARD 26 contains the collective pitch and rotor angular velocity arguments. The number of collective pitch values and rotor angular velocity values to be read is specified by NTHC and NPSID on Card 03, respectively. Should the collective pitch or rpm fall outside the tabulated domain, extrapolation will not be made. Instead, the program will use the closest value obtainable from the table, i.e. a value on the boundary of the tabulated domain.

CARD 27 is used to identify the first input blade mode, i.e. collective or cyclic, flatwise or edgewise etc.

CARD 28* contains the natural frequencies, in radians per second, for the first normal blade mode. The frequencies are read in first as a function of the collective (pitch for the lowest rpm), then for successively higher rpm.

*Indicates more than one card may be required for these data.

CARD 29 allows the user to specify if the shapes of the normal modes are functions of: (1) both rotor rpm and collective, (2) collective pitch only, and (3) rpm only.

If LC(L) = 1, means the mode shapes are functions of rotor rpm and collective pitch.

If LC(L) = 2, means the mode shapes are functions of collective pitch only.

If LC(L) = 3, means the mode shapes are functions of rpm only.

If LC(L) = 4, means the mode shapes do not vary with rpm or collective pitch.

There are two types of mode shapes associated with gimballed rotors, collective modes, and cyclic modes. The former are polar symmetric modes; the latter polar antisymmetric modes. KDR on Card 29 specifies whether the input mode is a collective or cyclic mode. KDR is used only if IT12 equals 0 or 2.

If KDR = 0, a collective mode is input.

If KDR = 1, a cyclic mode is input.

CARD 30* contains the matrix of out-of-plane, in-plane, and torsional components for the blade normal mode. The out-of-plane component is read in first, followed by the inplane and torsional components. Each component is read in from root to tip, first versus collective pitch, then versus rpm.

CARD 31* contains the matrix of beamwise and chordwise normalized bending moment for the first normal mode. The beamwise moments are read in first, followed by the chordwise moments. Each component is read in first versus collective pitch, then versus rpm. NMI on Card 03 specifies the number of blade stations where moment data will be read in.

CARDS 32* to 36*, 37* to 41*, and 42* to 46* are arranged in the same order as Cards 27 through 31. The data on cards 32 to 36 are for the second blade mode and should be deleted if NBM on Card 03 ≤ 1.

*Indicates more than one card may be required for these data.

The data on cards 37 to 41 are for the third blade mode and should be deleted if NBM on Card 03 ≤ 2. The data on cards 42 to 46 are for the fourth blade mode and are deleted if NBM on Card 03 ≤ 3.

CARD 47 contains the initial conditions for the normal blade modes. The initial conditions for the first mode of each blade are read in first, followed by those for the second blade mode, etc.

CARD 48 is used to specify parameters which indicate the amount of the input data to follow.

NSW is the number of wing lumped mass stations. 0 < NSW ≤ 10. NSW is read in only if IT5 = 1.

NSHP is the number of normal modes used for the wing/pylon system. 0 < NSHP ≤ 4. NSHP is read in only if IT5 = 1.

NPDD is the number of pylon accelerometers. 0 < NPDD ≤ 9. NPDD is used only if IT5 = 1 and IT10 = 1.

CARD 49 contains data for the wing/pylon system.

WZETA is the wing structural damping ratio. A zero value is acceptable.

PM is the lumped pylon mass (complete nacelle mass less the rotor group) PM ≥ 0.

ZP is the location (Z direction) of the pylon center of gravity along the shaft centerline from the conversion axis (see Figure 6).

YP is the lateral location (Y direction) of the pylon center of gravity with respect to the shaft axis (see Figure 6).

XP is the vertical location (X direction) of the pylon center of gravity with respect to the shaft axis (see Figure 6).

HM Concentrated hub mass (rotor group mass less the blade's mass) HM ≥ 0 (see Figure 6).

WEPE is the mean distance from the wing aerodynamic center to the wing elastic axis. Positive if the aerodynamic center is ahead of the elastic axis.

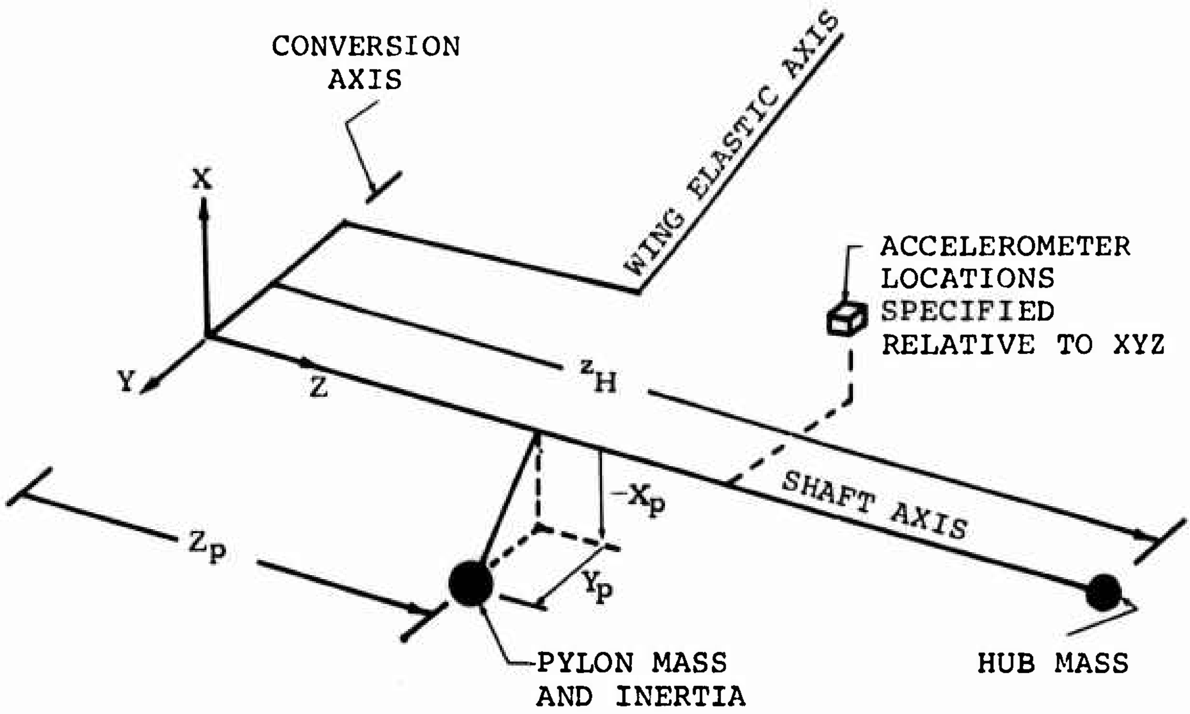


Figure 6. Pylon Geometries.

WIPP is the pylon pitching mass moment-of-inertia about the pylon center of gravity.

CARD 50 contains SHPE(J), the initial conditions for the wing normal modes (J = 1 to NSHP).

CARD 51* contains the chordwise components of the wing normal modes. They are read in from wing root to wing tip for first mode, then for the second mode, etc. The number of the discrete wing stations is specified by NSW on Card 48.

CARD 52* contains the beamwise components of the wing normal modes. They are read in the same order as the chordwise components.

*Indicates more than one card may be required for these data.

CARD 53* contains the torsional components of the wing normal modes. They are read in the same order as the chordwise components.

CARD 54* contains the wing lumped inertial properties. The masses from wing root to tip are read in first followed by the torsional mass moments-of-inertia about the wing elastic axis, and the wing mass unbalances. The wing segment widths are read in last. The number of discrete wing stations for which data must be supplied is specified by NSW on Card 48. Wing mass unbalances are considered positive if wing center of gravities are in front of the wing elastic axis. Wing aerodynamic segment lengths can be set to zero in order to zero out wing aerodynamic forces.

CARD 55* contains the natural frequencies and pylon mode shapes for the wing normal modes. The number of modes is specified by NSHP on Card 48. The natural frequencies are read in first, in succession for each mode followed by the normalized pylon fore and aft deflections, the normalized pylon lateral deflections, the normalized vertical deflections, the normalized pylon yaw angles, the normalized pylon pitch angles, and the wing/pylon effective masses. The wing/pylon effective masses will be calculated by the program if IT13 = 0. In this case, the input values for the wing/pylon effective mass are not used.

CARD 56* contains a modified wing stiffness matrix by rows. The wing moments are calculated using the relationship between force and normal coordinates given by:

$$\{F\} = [k]\ \{x\}$$

in which $\{F\}$ is the force matrix, $\{x\}$ the normal coordinates matrix, and $[k]$ the modified wing stiffness matrix obtained by premultiplying the mode shape matrix by a stiffness matrix. The order of $[k]$ is determined by the number of wing normal modes, i.e. the order of $[k]$ = 10 x NSHP on Card 48.

*Indicates more than one card may be required for these data.

CARD 57* is used to specify the location of up to nine (9) accelerometers located on the pylon. The locations are referred to the pylon coordinate system. The X, Y, Z components are read in first, followed sequentially by those for the second accelerometer etc. The number of accelerometers is specified by NPDD on Card 48.

3. Output Format and Guide

The output of ARAP06 consists of two parts: (1) the input data and (2) a time history of the system displacements, accelerations, and loads. Machine (CALCOMP) plots of selected variables may be obtained if desired.

The print-out of the input data is in order of the flight condition being simulated, options, proprotor data, and wing/pylon data. All of the data used in the case is printed out.

The printed time history output may be conveniently divided into seven groups. The first group, consisting of the first two lines, is always printed. The printing of the other groups depends upon the options selected.

a. Proprotor Rigid-Body Motion and Forces and Moments (Rows 1 and 2)

TIME is the real time (sec)

PSID is the rotor instantaneous rotational speed about the rotor shaft axis (rpm)

PSI is the azimuth position of the master blade, measured from X-axis of the rotor-fixed system (deg)

BET-LG is the longitudinal flapping in rotor-fixed system positive about negative Y-axis (deg)

BET-LT is the lateral flapping in rotor-fixed system, positive about positive X-axis (deg)

BET(1) is the total flapping of the master blade, positive if flapping up (deg)

FFX is the rotor aerodynamic H-force referred to rotor-fixed system. Positive along positive X-axis (lb)

*Indicates more than one card may be required for these data.

FFY is the rotor aerodynamic side force referred to rotor-fixed system. Positive along positive **Y-axis** (lb)

FFZ is the rotor aerodynamic drag force, referred to rotor-fixed system. Positive along positive **Z-axis** (lb)

WBX is the **X-component** of the gravitational force on rotor-blades, referred to rotor-fixed system. Positive along positive **X-axis** (lb)

WBY is the **Y-component** of the gravitational force on rotor-blades, referred to rotor-fixed system. Positive along positive **Y-axis** (lb)

WBZ is the **Z-component** of the gravitational force on rotor-blades, referred to rotor-fixed system. Positive along positive **Z-axis** (lb)

FI is the component of rotor inertial loads in rotor-rotating system, positive along **x-axis (lb)**

FJ is the component of rotor inertial loads in rotor-rotating system, positive along **y-axis (lb)**

FK is the component of rotor inertial loads in rotor-rotating system, positive along **z-axis (lb)**

MI is the component of rotor inertial moments in rotor-rotating system, positive about **x-axis (in-lb)**

MJ is the component of rotor inertial moments in rotor-rotating system, positive about **y-axis (in-lb)**

MK is the component of rotor inertial moments in rotor-rotating system, positive about **z-axis (in-lb)**

TFFX is the component of net rotor force referred to rotor-fixed system. Positive along **X-axis** (lb). And

$$TFFX = FFX + FI \cos\psi - FJ \sin\psi + WBX$$

TFFY is the component of net rotor force referred to rotor fixed system. Positive along **Y-axis** (lb). And

$$TFFY = FFY + FI \sin\psi + FJ \cos\psi + WBY$$

TFFZ is the component of net rotor force referred to rotor fixed system. Positive along **Z-axis** (lb). And

$$TFFZ = FFZ + FK + WBZ$$

TFMX is the component of net rotor moment referred to rotor fixed system. Positive about **X-axis** (in-lb)

TFMY is the component of net rotor moment referred to rotor fixed system. Positive about **Y-axis** (in-lb)

TFMZ is the component of net rotor moment referred to rotor fixed system. Positive about **Z-axis** (in-lb)

b. Blade Deflections and Bending Moments

This group is calculated and printed only if IT4 = 0. The bending moments are written symbolically as follows:

JBMN where BM stands for a beamwise moment, J stands for the Jth blade, N stands for the Nth blade station at which the moment is calculated (in-lb)

JCMN where CM stands for chordwise moments, J and N have been defined above (in-lb)

The blade tip deflections are with respect to the preconed axis, and are written symbolically as follows:

ZTIPJ where TIP stands for blade tip, J stands for the Jth blade, Z indicates a beamwise deflection. It is positive if the blade is being bent upward. (in)

XTIPJ where X indicates a chordwise deflection. It is positive in the lag sence, i.e. compression in trailing edge (in)

The normal coordinates for the blade normal modes are written symbolically as follows:

JQBN where QB stands for the normal coordinates, J indicates the Jth blade, N stands for the Nth normal mode

c. Deflections and Rotations of Wing/Pylon System

This group is executed and printed only if IT5 = 1.

THP is the pylon instantaneous pitch angle, positive about Y_F-axis in the inertial fixed system. THP contains the pylon steady conversion angle as well as the pitch due to wing torsional deflections (rad)

PSIP is the pylon instantaneous yaw angle, positive about Z_F-axis in the inertial fixed system (rad)

XHB is the X_F-component of hub deflection, referred to reference inertial coordinate system (in)

$$X_{HB} = X_o + z_H(\cos\psi_p \cos\theta_p - 1)$$

in which X_o is X_F-component deflection of pylon conversion axis, z_H is the total pylon length (CA to hub), ψ_p is the pylon yawing angle, and θ_p is the pylon pitching angle

YHB is the Y_F-component of hub deflection, referred to reference inertial coordinate system (in)

$$Y_{HB} = Y_o + z_H \sin\psi_p$$

ZHB is the Z_F-component of hub deflection, referred to reference inertial coordinate system (in)

$$Z_{HB} = Z_o - z_H \cos\psi_p \sin\theta_p$$

WBM is the beamwise deflection at the wing-tip, positive down (in)

WCD is the chordwise deflection at the wing-tip, positive forward (in)

WTN	is the torsional deflection at the wing-tip, positive leading-edge up (rad)
QW1	is the first normal coordinate of the wing/pylon system
QW2 QW3 QW4	Second, third, and fourth normal coordinates of the wing/pylon system, respectively

d. Wing Loads and Moments

The execution and printing of this group are controlled by option IT11. If IT11 = 1, the following variables are printed.

FXL	is the chordwise shearing force at the left end of a wing segment, positive forward (lb)
FZL	is the beamwise shearing force at the left end of a wing segment, positive down (lb)
MXL	is the beamwise bending moment at the left end of a wing segment, positive if tension is produced on the wing upper surface (in-lb)
MYL	is the torsional moment at the left end of a wing segment, positive if wing leading edge is up (in-lb)
MZL	is the chordwise bending moment at the left end of a wing segment, positive if tension is produced on the wing surface at the leading-edge (in-lb)

The remaining variables in this group are chordwise and beamwise shearing forces, beamwise, torsional, and chordwise bending moments for the right end of the wing segment.

e. Pylon Accelerations

This group is controlled by option IT10. If IT10 = 1, the computation of pylon accelerations is executed.

XPDD1	is the X-component acceleration of the first accelerometer, referred to pylon coordinate system (in fraction of g)

YPDD1	is the Y-component acceleration of the first accelerometer, referred to pylon coordinate system (in fraction of g)
ZPDD1	is the Z-component acceleration of the first accelerometer, referred to pylon coordinate system (in fraction of g)

The remaining variables in this group are component accelerations for the accelerometers, --NPDD.

f. Peak Values of Selected Variables

The printing-out of this group is controlled by option IT9. If IT9 = 1, peaks of master blade flapping, wing-tip beamwise deflection, wing-tip chordwise deflection, and first normal coordinate of wing/pylon are selected versus time. They are printed at the end of each run.

g. Details of Rotor and Wing Aerodynamics

This group is printed at the end of the second rotor revolution for all runs. Instantaneous aerodynamic details at all blade and wing stations are printed for reference.

(1) Rotor Airloads Details (see Figure 7)

VP	is the vertical component of the blade local velocity (in/sec)
VT	is the tangential component of the blade local velocity (in/sec)

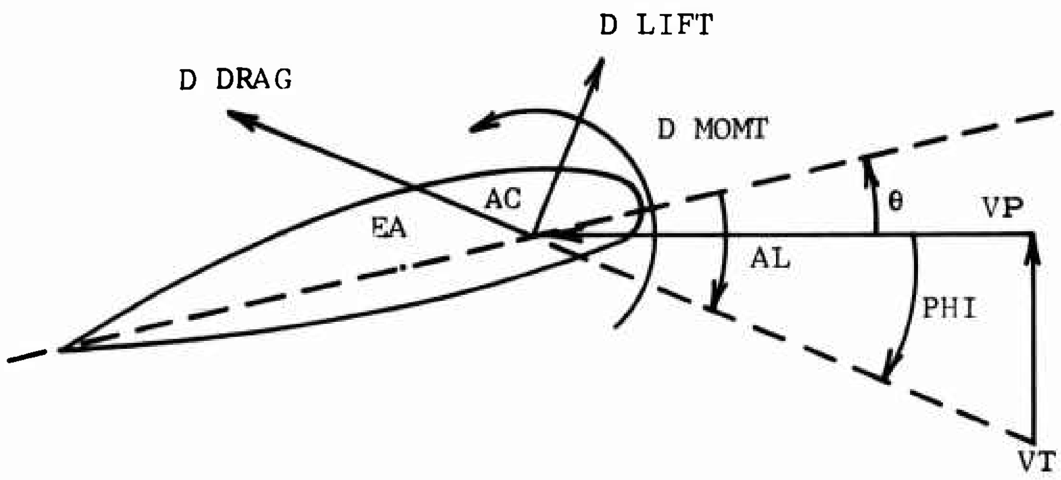


Figure 7. Blade Element Aerodynamic Details.

PHI — is the inflow angle (rad)

AL — is the angle of attack (rad)

CL,CD CM — is the local blade lift, drag, and pitching moment coefficients respectively

MACH — is the local Mach number

DHF — is the rotor in-plane aerodynamic drag force, it is positive if it tends to slow down the rotor (lb)

DQ — is the aerodynamic pitching moment about the blade elastic axis, positive for leading edge nose up (lb)

DRADF — is the blade aerodynamic radial force in shaft plane. Positive if it points to the blade root (lb)

DTHST — is the blade aerodynamic thrust perpendicular to shaft plane. Positive if it is outward from the shaft plane (lb)

DLIFT — is the local aerodynamic lift at the quarter chord. It is perpendicular to the local total velocity and is calculated directly from C_L (lb)

DDRAG — is the blade local aerodynamic drag at the quarter chord. It is parallel to the local total velocity and is calculated directly from C_D (lb)

DMOMT — is the blade local aerodynamic pitching moment at the quarter chord. It is positive if blade leading-edge is up, and is calculated directly from C_M (in-lb)

(2) Wing Airloads Details (see Figure 8)

VP — is the vertical component of the wing local velocity (in/sec)

VT — is the tangential component of the wing local velocity (in/sec)

PH — is the local inflow angle (rad)

AL — is the local angle of attack (rad)

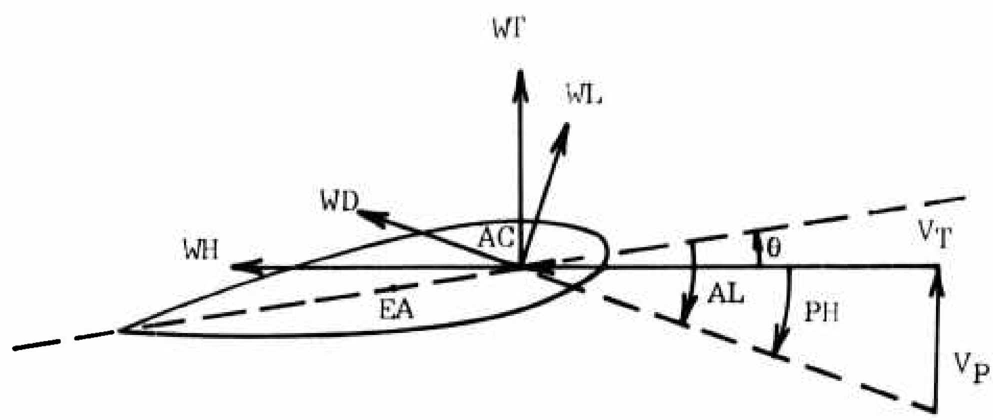


Figure 8. Wing Element Aerodynamic Details.

CL is the local lift coefficient for the wing, calculated by

$$C_L = \frac{\partial L}{\partial \alpha} \cdot \alpha$$

where

$\partial L/\partial \alpha$ is the lift curve slope, an input on Card 11 and, α is the local angle of attack

CD is the local drag coefficient

WL is the local aerodynamic lift force

WD is the aerodynamic drag force

WT is the local thrust force perpendicular to the free stream (lb)

WH is the wing local drag force parallel to the free stream (lb)

WQ is the local torsional moment about the elastic axis, positive leading-edge is up (in-lb)

The machine plots are self-explanatory. An example is given in Appendix I.

4. Suggested User Techniques

a. Data Deck Build-up

In preparing a data deck for program ARAP06, the following steps are suggested.

(1) Start With Rotor Rigid-Body Freedom - The user should start with rotor rigid-body equations (OPTION IT5 = 0). The data required for this option consist of the blades inertia and aerodynamic properties, i.e. the blade twist, mass distribution, mass unbalance, variation of blade chord with blade radius, rotor precone, pitch-flap coupling, and blade collective pitch. Tabulated forms for the blade section C_L, C_D, and C_M are also needed. These data form a fundamental data deck for the program which can be readily checked out.

(2) Add Wing Elastic Degrees-of-Freedom - After checking out the rotor rigid-body degrees-of-freedom, the wing flexible motion should be added (IT5 = 1). This requires that the wing/pylon inertial properties, linear dimensions and normal modes as well as natural frequencies be supplied. The lift-curve slope of wing airfoil must also be estimated. The wing equations should first be checked out with the proprotor blade segment lengths set to zero which eliminate the proprotor aerodynamics. Checkout should include verification of coupled mode shapes and frequencies.

(3) Add Blade Elastic Degrees-of-Freedom - The proprotor blade elastic motion should be added last. This requires that the blade natural frequencies, normal modes, and normalized blade bending moments for each mode be supplied in tabulated forms (IT4 should be 0).

(4) Other Options - The various ARAP06 options such as those for wing/rotor aerodynamic interference, blade feathering or gust encounter should be used only after the basic math model has been checked out. Many of the options require hand calculation prior to their use. For example, if option IT2 is to be used to feather or unfeather the proprotor, the proper blade collective pitches should be calculated by equation 134 in Volume I. Likewise for the 1-cosine gust encounter option, IT3 = 2, the gust encounter period may be calculated by

$$\Delta t = \frac{25\ WC}{U}$$

in which WC is the wing mean aerodynamic chord, U is the velocity along flight path. These are input parameters on card 11 and card 05, respectively.

b. Machine Time Requirements

The ratio of the machine time to the simulated real time depends upon flight conditions and options. Since the equations of motion in ARAP06 are nonlinear in nature and are solved simultaneously by iterations, the more complicated the input condition the greater the number of iterations required. A few examples of machine time requirements are given below. (The machine time here referred is based upon an IBM 360-65.)

Example 1 - The blade rigid-body flapping and rotor angular freedom are locked out. The blades are inelastic and the proprotor is coupled with the wing/pylon degrees-of-freedom (4 modes). Forty seconds of machine time for one second of real time was required.

Example 2 - With the blade flapping and rotor angular freedoms added, the machine time is increased to ninety seconds per secound of real time.

Example 3 - With the blade rigid-body flapping and rotor angular freedoms locked out, but with flexible blades (4 modes) coupled with wing/pylon motions (4 modes), approximately six minutes of machine time is required for one second of real time.

Example 4 - If the proprotor blade flapping and angular freedom is added to example 3, approximately nine minutes of machine time is required for one second of real time.

Proper choice of itegration step has impact on both the machine time and program accuracy. Too small an integration step increases the machine time. However, too large integration step will prevent the program from giving accurate results. The user should experiment with integration step and establish the largest, accurate step.

c. Acceleration Averaging Requirements

Numerical difficulties are occasionally encountered with ARAP06 because of the nonlinearity of the equations of motion. This manifests itself as a higher frequency vibration at the time step frequency. A weighted average of the accelerations can be used to eliminate this problem. This option is controlled by IT14.

B. DFAL17 - FPR Blade Folding Flutter Analysis

The inputs to DFAL17 consist of the geometry, mass and stiffness distributions of the wing, pylon, and blades. The following section properties can vary with blade fold angle: strip width, chord, reference semispan and chord, cosine of blade fold angle, mean aerodynamic chord, distance between dumbbells, blade sweep, distance from elastic axis to dumbbells, mass unbalance, and inertia.

The card order in a DFAL17 data deck is shown in Figure 9. If the program is being used to calculate natural frequencies and mode shapes, the aerodynamic data set is not required.

1. Input Format

CARD 01 Format 20A4:

Title Card; Column 1-80, alphameric.

CARD 02 Format I4:

Col. 1 - 4	NSIZ	Matrix size = 74, if minus, punch flexibility matrix and eigenvalues
Col. 5 - 8	NAERO	Reduced velocities (1/K), if NAERO = 0, skip Cards 3-19 (aero input)
Col. 9 - 12	MODES	Number of eigenvalues to be calculated

Format E12.5:

Col. 13 - 24	FK	Flex scaling factor; if FK = 0 then a value of FK = 1 is assumed
Col. 25 - 36	AMK	Mass scaling factor; if AMK = 0, FM = 1
Col. 37 - 48	RHO	Density; if RHO = 0, (slug/ft^3) RHO = 0.002378

Cards 03 through 19 contain aerodynamic inputs, omit Cards 03 through 19 if NAERO = 0.

CARD 03 Format E12.5:

Col. 1 - 12	BR	Reference semichord	(ft)
Col. 13 - 24	s	Reference semispan	(ft)
Col. 25 - 36	BREF	Reference semichord	(ft)
Col. 37 - 48		Reduced velocities in	
Col. 49 - 60	VELCTY	fields of E12.5 until	
Col. 61 - 72		NAERO values have been entered	

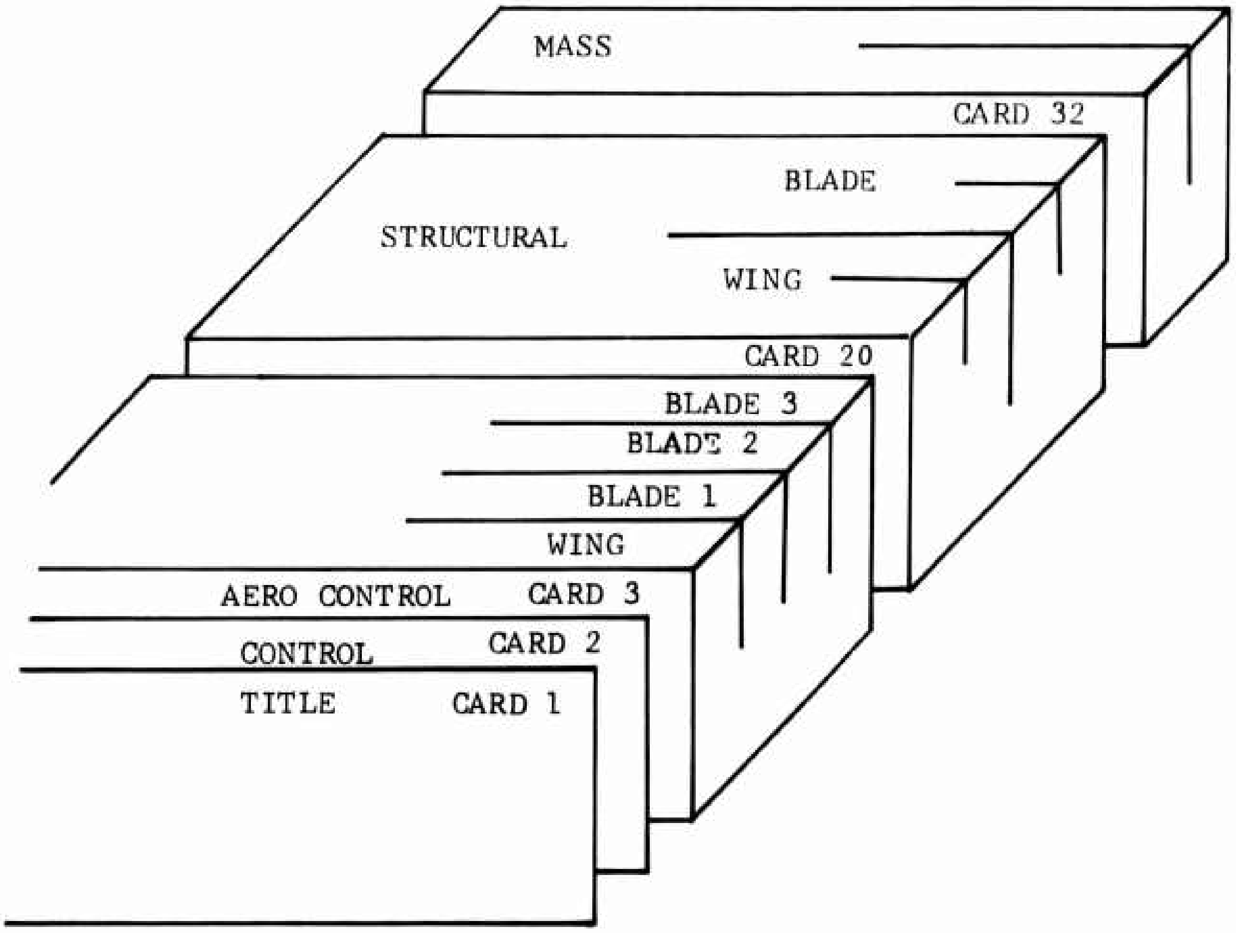


Figure 9. Card Order in DFAL17 Data Deck.

CARD 04	Format E12.5:			
	Col. 1 - 12	COSLMA	Cosine of wing sweep angle	
	Col. 13 - 24	BRA	Reference wing semichord	(ft)
	Col. 25 - 36	SA	Reference wing semispan	(ft)
	Col. 37 - 48	CAPA	Reference wing area	(ft^2)
	Col. 49 - 60	CBAA	Wing mean aerodynamic chord	(ft)
CARD 05*	Format 6E12.5:			
	Col. 1 - 72	$DELTAY_i$	Width of wing panels (5 wing, 2 pylon)	(ft)
	Format E12.5:			
	Col. 1 - 12			
CARD 06*	Format 6E12.5:			
	Col. 1 - 72	B_i	Wing panel semichords (5 wing, 2 pylon)	(ft)
	Format E12.5:			
	Col. 1 - 12			
CARD 07*	Format 6E12.5:			
	Col. 1 - 72	D_i	Distance between dumbbells on wing & pylon (5 wing, 2 pylon)	(ft)
	Format E12.5:			
	Col. 1 - 12			
CARD 08	Format E12.5:			
	Col. 1 - 12	COSLMA	Cosine of blade fold angle	
	Col. 13 - 24	BRA	Reference semichord	(ft)
	Col. 25 - 36	SA	Reference semispan	(ft)
	Col. 37 - 48	CAPA	Reference area	(ft^2)
	Col. 49 - 60	CBAA	Mean aerodynamic chord	(ft)
CARD 09	Format 5E12.5:			
	Col. 1 - 60	$DELTAY_i$	Blade 1, section widths	(ft)
CARD 10	Format 5E12.5:			
	Col. 1 - 60	B_i	Blade 1, section semichords	(ft)
CARD 11	Format 5E12.5:			
	Col. 1 - 60	D_i	Blade 1, distance between dumbbells	(ft)

*Indicates more than one card may be required for these data.

CARD 12	Format E12.5:				
	Col. 1 - 12	COSLMA	Cosine of blade fold angle	
	Col. 13 - 24	BRA	Reference semichord	(ft)
	Col. 25 - 36	SA	Reference semispan	(ft)
	Col. 37 - 48	CAPA	Reference area	(ft^2)
	Col. 49 - 60	CBAA	Mean aerodynamic chord	(ft)
CARD 13	Format 5E12.5:			
	Col. 1 - 60	$DELTAY_i$	Blade 2, section widths	(ft)
CARD 14	Format 5E12.5:			
	Col. 1 - 60	B_i	Blade 2, section semichords	(ft)
CARD 15	Format 5E12.5:			
	Col. 1 - 60	D_i	Blade 2, distance between dumbbells	(ft)
CARD 16	Format E12.5:			
	Col. 1 - 12	COSLMA	Cosine of blade sweep angle	
	Col. 13 - 24	BRA	Reference semichord	(ft)
	Col. 25 - 36	SA	Reference semispan	(ft)
	Col. 37 - 48	CAPA	Reference area	(ft^2)
	Col. 49 - 60	CBAA	Mean aerodynamic chord	(ft)
CARD 17	Format 5E12.5:			
	Col. 1 - 60	$DELTAY_i$	Blade 3, section widths	(ft)
CARD 18	Format 5E12.5:			
	Col. 1 - 60	B_i	Blade 3, section semichords	(ft)
CARD 19	Format 5E12.5:			
	Col. 1 - 60	D_i	Blade 3, distance between dumbbells	(ft)

<u>Structural Input</u>

CARD 20	Format E12.5:			
	Col. 1 - 12	WGSWEP	Wing sweep, forward is minus	(deg)
	Col. 13 - 24	BLDSWP	Blade sweep, aft is plus	(deg)

Col. 25 - 36	VRPYLS	Pylon fore and aft stiffness	(in-lb/rad)
Col. 37 - 48	LTPYLS	Pylon lateral stiffness	(in-lb/rad)
Col. 49 - 60	FOLDSF	Blade fold stiffness	(in-lb/rad)
Col. 61 - 72	SMASTF	Mast torsional stiffness	(in-lb/rad)

CARD 21 Format E12.5:

Col. 1 - 12	FXCLST	Fixed control stiffness	(in-lb/rad)
Col. 13 - 24	MVCLST	Movable control stiffness	(in-lb/rad)
Col. 25 - 36	LCA	Distance elastic axis to conversion axis	(in)
Col. 37 - 48	PYLOFF	Pylon offset	(in)
Col. 49 - 60	H	Mast length	(in)
Col. 61 - 72	EHUB	Hub offset	

CARD 22 Format E12.5:

Col. 1 - 12	LW	Length wing segment 1	(in)
Col. 13 - 24	EIBW	EI beam wing segment 1	($lb-in^2$)
Col. 25 - 36	EICW	EI chord wing segment 1	($lb-in^2$)
Col. 37 - 48	GJW	CG wing segment 1	($lb-in^2$)

CARD 23 Format E12.5:

Col. 1 - 12	LW	Length wing segment 2	(in)
Col. 13 - 24	EIBW	EI beam wing segment 2	($lb-in^2$)
Col. 25 - 36	EICW	EI chord wing segment 2	($lb-in^2$)
Col. 37 - 48	GJW	GJ wing segment 2	($lb-in^2$)

CARD 24 Format E12.5:

Col. 1 - 12	LW	Length wing segment 3	(in)
Col. 13 - 24	EIBW	EI beam wing segment 3	($lb-in^2$)
Col. 25 - 36	EICW	EI chord wing segment 3	($lb-in^2$)
Col. 37 - 48	GJW	GJ wing segment 3	($lb-in^2$)

CARD 25	Format E12.5:			
	Col. 1 - 12	LW	Length wing segment 4	(in)
	Col. 13 - 24	EIBW	EI beam wing segment 4	(lb-in^2)
	Col. 25 - 36	EICW	EI chord wing segment 4	(lb-in^2)
	Col. 37 - 48	GJW	GJ wing segment 4	(lb-in^2)
CARD 26	Format E12.5:			
	Col. 1 - 12	LW	Length wing segment 5	(in)
	Col. 13 - 24	EIBW	EI beam wing segment 5	(lb-in^2)
	Col. 25 - 36	EICW	EI chord wing segment 5	(lb-in^2)
	Col. 37 - 48	GJW	GJ wing segment 5	(lb-in^2)
CARD 27	Format E12.5:			
	Col. 1 - 12	LB	Length blade segment 1	(in)
	Col. 13 - 24	EIBB	EI beam blade segment 1	(lb-in^2)
	Col. 25 - 36	EIBC	EI chord blade segment 1	(lb-in^2)
	Col. 37 - 48	GJB	GJ blade segment 1	(lb-in^2)
CARD 28	Format E12.5:			
	Col. 1 - 12	LB	Length blade segment 2	(in)
	Col. 13 - 24	EIBB	EI beam blade segment 2	(lb-in^2)
	Col. 25 - 36	EIBC	EI chord blade segment 2	(lb-in^2)
	Col. 37 - 48	GJ	GJ blade segment 2	(lb-in^2)
CARD 29	Format E12.5:			
	Col. 1 - 12	LB	Length block segment 3	(in)
	Col. 13 - 24	EIBB	EI beam blade segment 3	(lb-in^2)
	Col. 25 - 36	EIBC	EI chord blade segment 3	(lb-in^2)
	Col. 37 - 48	GJB	GJ blade segment 3	(lb-in^2)
CARD 30	Format E12.5:			
	Col. 1 - 12	LB	Length blade segment 4	(in)

	Col. 13 - 24	EIBB	EI beam blade segment 4	($lb\text{-}in^2$)
	Col. 25 - 36	EIBC	EI chord blade segment 4	($lb\text{-}in^2$)
	Col. 37 - 48	GJB	GJ blade segment 4	($lb\text{-}in^2$)
CARD 31	Format E12.5:			
	Col. 1 - 12	LB	Length blade segment 5	(in)
	Col. 13 - 24	EIBB	EI beam blade segment 5	($lb\text{-}in^2$)
	Col. 25 - 36	EIBC	EI chord blade segment 5	($lb\text{-}in^2$)
	Col. 37 - 48	GJB	GJ blade segment 5	($lb\text{-}in^2$)

<u>Mass Input</u>

CARDS 32 & 33	Format 6E12.5:			
	Col. 1 - 72	X_i	Distance EA to forward central point--12 total, 5 wing, 2 pylon, 5 blade is positive	(in)
CARDS 34 & 35	Format 6E12.5:			
	Col. 1 - 72	Y_i	Distance EA to aft central point--12 total, 5 wing, 2 pylon, 5 blade is positive	(in)
CARDS 36 & 37	Format 6E12.5:			
	Col. 1 - 72	M	Weight to be lumped at each dumbbell	(lb)
CARDS 38 & 39	Format 6E12.5:			
	Col. 1 - 72	S	Unbalance to be lumped at dumbbell is minus if forward of EA	(in-lb)
CARDS 40 & 41	Format 6E12.5:			
	Col. 1 - 72	I	Inertia to be lumped at each dumbbell	($in\text{-}lb^2$)

2. Guide to Input Format

CARD 01	TTL	is used to identify case being run. All alphabetic or numeric information is reproduced and centered on first output page.
CARD 02	NSIZ	is the matrix size (74 max). If this number is negative (-74), then the structural influence coefficient matrix and all 74 eigenvalues are punched on cards (6E12.5 format) for use as input to DFAL18. The flexibility matrix and eigenvalues are punched for the first 1/K.
	NAERO	is the number of reduced velocities (1/K) $\leq$ 15. If NAERO = 0, cards 3 through 19 are omitted. If NAERO is negative, the aerodynamic matrix is printed out.
	MODES	is the number of eigenvalues to be printed out and the number of eigenvectors to be calculated.
	FK	is the flexibility (SIC) matrix scaling factor. The EI and GI will be multiplied by this factor to reduce input data requirements. It can also be used for quick stiffness changes in parametric studies. If FK = 0 or blank, a 1 is assumed.
	AMK	is the mass matrix scaling factor. The section weights, inertias, and unbalances are multiplied by AMK. This can be used for quick mass changes in parameter changes. If AMK = 0 or blank, a 1 is assumed.
	RHO	is the air density (slug/ft^3). It is used in the calculation of airloads. If RHO = 0 then 0.002378 is assumed.
CARD 03*	BR	is the reference semichord (ft). The calculated aerodynamics are multiplied by BR to form oscillatory airloads.
	S	is the reference semispan (ft). The calculated aerodynamics are multiplied by S to form oscillatory airloads.
	BREF	The reference semichord (ft). Usually the same as BR. Is used only in the velocity calculation

*Indicates more than one card may be required for these data.

$$V = \frac{1}{K} * \frac{BREF}{\lambda_R}$$

where λ_R = real part of the eigenvalue.

VELCTY is the reduced velocity ($1/K = \frac{V}{B_R\omega}$) values. There will be three on Card 3 and six on each succeeding card until NAERO reduced velocities are input.

CARD 04 COSLMA is the cosine of the wing sweep angle.

BRA is the reference wing semichord (ft). This should be same as BR on Card 3.

SA is the reference wing semispan (ft). This should be the same as S on Card 03.

CAPA is the reference area (ft^2). This should be the total area of the wing and three blades.

CBAA is the wing mean aerodynamic chord (ft).

$$CBAA = \frac{2}{3} C_T \left[e + \frac{1}{e+1} \right]$$

where $e = \frac{\text{chord root}}{\text{chord tip}}$

C_T = chord tip

CARD 05* $DELTAY_i$ is delta y (the strip width of the wing segments. There are seven strips--five for the wing and two representing the pylon. Strip six is for pylon pitch airloads and strip seven for pylon yawing airloads. These values can be scaled for 3-D lift distribution using the formula

$$DY_i = \frac{a_i}{2\pi} \cdot D\overline{Y}_i$$

where a_i is the estimated lift curve slope of the individual segment (sweep effects neglected), $D\overline{Y}_i$ is the actual geometric strip width.

Two cards are required since there are seven strips. $D\overline{Y}_i$ can be zero if desired.

CARD 06	B_i	is the wing section semichord (ft). These are the semichords for each wing segment plus the two pylon sections. Two cards are required. B_i must be nonzero.
CARD 07	D_i	is the distance (ft) between the forward and aft dumbbell. Normally the forward control point is on the quarter chordline and the aft control point is on the three-quarter chordline. This makes D_i equal to B_i. D_i must be nonzero.
CARDS 08, 09, 10, 11	(Blade 1 aero data)	same format as cards 04 through 07 except blade parameters are used. There are only 5 sections for each blade, so there are only 5 DY_i, 5 B_i, and 5 D_i.
CARDS 12, 13, 14, 15	(Blade 2 aero data)	same format as cards 08 through 11.
CARDS 16, 17, 18, 19	(Blade 3 aero data)	same format as cards 08 through 11.
CARD 20	WGSWEP	is the wing sweep angle (degrees). Forward sweep is negative.
	BLDSWP	is the blade fold angle (degrees). Fold aft is positive. Precone is a negative blade fold angle.
	VRPYLS	is the pylon fore and aft (pitch) mounting spring rate (in-lb/rad). This number is used to calculate the GJ of the pylon offset element.
	LTPYLS	is the pylon lateral mounting spring rate (in-lb/rad). This stiffness is used to calculate the chord EI of the pylon offset element.
	FOLDSF	is the blade fold hinge spring rate (in-lb/rad). This stiffness is used to calculate the EIC of the one-inch element located between the fold hinge and the first blade element.
	SMASTF	is the mast torsional stiffness (in-lb/rad). This number is used to calculate the mast GJ.

CARD 21

FXCLST — is the fixed pitch change axis spring rate (in-lb/rad). This stiffness is used to calculate the GJ of the element inboard of the fold hinge.

MVCLST — is the moveable pitch change axis spring rate (in-lb/rad). This stiffness is used to calculate the GJ of the one-inch element located between the blade fold axis and the first blade segment.

LCA — is the distance in inches between the wing elastic axis and the conversion axis. Parallel to rotor shaft. Positive if conversion axis is aft of the elastic axis.

PYLOFF — is the lateral distance from the wing tip to the shaft centerline (pylon offset).

H — is the distance from the intersection of the conversion axis and the shaft axis to the hub (mast length - in).

EHUB — is the distance from the intersection of the shaft centerline and hub to the blade fold axis (fold hinge offset - in).

CARDS 22-26 — contain wing structural data. Each card corresponds to a wing segment. The format for each card is the same.

LW — is the length of ith wing segment in inches

EIBW — is the beamwise EI for ith wing segment in lb-in^2

EICW — is the chordwise EI for ith wing segment in lb-in^2

GJW — is the GJ for ith wing segment in lb-in^2

Each wing structural segment ends at a lumped mass point. The mass point should be located in the center of the corresponding aerodynamic panel. The distance from the end of the fifth segment to the wing tip (segment 6) is assumed to be one-half the length of the fifth wing segment. The stiffness of segment 6 is assumed to be the same as segment 5.

CARDS 27-31 contain blade structural data. Each card corresponds to a blade segment. The blades are assumed to be identical. The format for each card is the same.

LB — is the length of ith blade segment in inches

EIBB — is the beamwise EI for ith blade segment in lb-in^2

EICB — is the chordwise EI for ith blade segment in lb-in^2

GJB — is the GJ for ith blade segment in lb-in^2

Each blade segment ends at a mass point which should be located at the center of the corresponding aerodynamic panel. Therefore the fifth blade segment will end short of the blade tip. The blade begins one-inch outboard of the fold hinge.

CARDS 32-41 contain inertial data for the wing and blades.

X_i — is the distance in inches from the elastic axis to the forward control point. The forward central point is located on the quarter chordline. X_i is used in the generation of the coupled mass matrix and the SIC matrix. X_i is always positive and it is advisable (for matrix conditioning) that X_i be of some value rather than zero. The X_i for the pylon could be the same as the mast length and thus show the hub motion. The five wing X_i, the pylon pitch, pylon yaw, and five X_i for one blade are input consecutively. The blades are assumed to be identical so the blade input is not repeated.

Y_i — is the distance in inches from the elastic axis to the aft control point. The aft control point is normally located on the three-quarter chordline. Y_i is used in the generation of the coupled mass matrix and the SIC matrix. Y_i is used in the generation of the coupled mass matrix and the SIC matrix. Y_i is always positive and it is advisable that it be of some value rather than zero. The five wing Y_i, the pylon pitch, pylon yaw, and the Y_i for one blade are input consecutively. The blades are assumed to be identical so the blade data are not repeated.

M — is the section weight in pounds. This weight is distributed on the fore and aft control points forming a dumbbell. M cannot be zero. M for the five wing segments, pylon pitch and yaw, and the five blade segments are in consecutive order. The hub weight is normally lumped in the pylon weight rather than the first blade segment.

S — is the section unbalance in in-lb. The unbalance is used in forming the mass matrix. A section center of gravity forward of the elastic axis produces a negative unbalance. S can be zero. The unbalance for the five wing segments, pylon pitch and yaw, and the five blade segments are in consecutive order.

I — is the section inertia I, in lb-in^2. This term is the inertia about the elastic axis. This parameter cannot be zero. The inertias for five wing segments, pylon pitch inertia, pylon yaw inertia, and five blade segments are entered consecutively.

3. Output Format and Guide

The output of DFAL17 consists of four parts:

- All input data.
- Constants which are internally calculated. These include the wing-tip spring rates, the flexibility and mass matrices, and the aerodynamic matrix, if desired.
- Flutter information consisting of the eigenvalues and vectors for each mode, the frequency structural damping required, and velocity in knots associated with each frequency.
- Punched cards containing the structural influence coefficient matrix and eigenvalues, if desired.

The output format is illustrated by a sample case in Appendix I.

4. Suggested User Techniques

a. Problems Encountered

Any large computer program is susceptible to numerical inaccuracies. DFAL17 is no exception. Because of the manipulation of large complex matrices, roundoff errors will occur. The numerical difficulties have been eliminated as much as possible, but the remaining oddities still remain.

- The tip spring rates (beam and chord--printed output) will be different for various blade fold angles. This variation is small (3-5 percent) and produces little change in output (frequency, damping, velocity, and mode shapes). The variation is caused by numerical inaccuracies during the condensation step.

- At higher reduced velocities, numerous modes will have printed "Imaginary frequency" in lieu of a frequency. Physically this signifies a divergence mode. By taking smaller reduced velocity increments prior to this point, the velocities, damping, and frequencies of interest can be calculated.

- All data must be checked for accuracy since input errors may produce little or no change in frequency but gross differences in damping required and mode shapes.

- When making an element "rigid" do not input EI or GJ more than 10^6 larger then the stiffnesses of surrounding elements. This can cause extreme numerical errors.

b. Usage Hints

Given below are some hints to make the program more flexible.

- Aerodynamics, as calculated by the program, are 2-D and incompressible. By variation of strip widths, semichords, and density, the effect of 3-D theory, compressibility, and altitude variations can be approximated. (See 3-D theory example, Figure 10.)

- Make full use of the flexibility and mass constants, the spring rate inputs, and pylon parameters. These terms will enable most parameter sweeps to be accomplished by single number change each run.

Example, 3-D theory:

The surface is assumed to have an elliptic lift distribution as shown below:

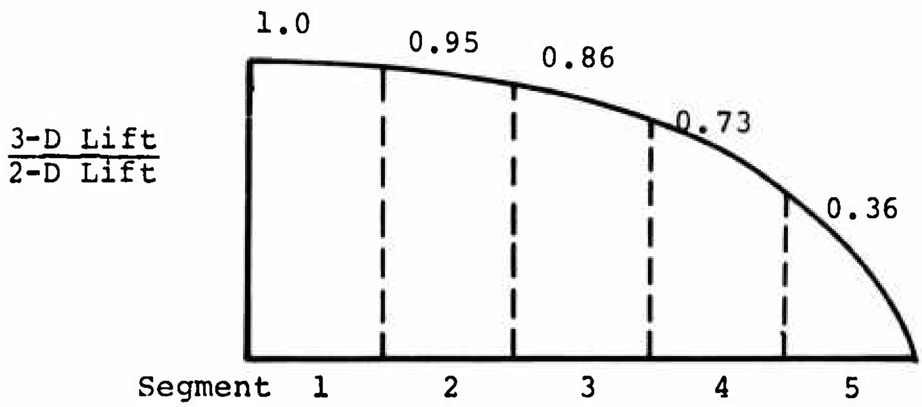


Figure 10. 3-D Lift Distribution.

To approximate 3-D theory, multiply the corresponding ΔY's by 1.0, 0.95, 0.86, 0.73, and 0.36.

To approximate the effect of compressibility, multiply the b_i's by the ratio of $\frac{C_L \text{ compressible}}{C_L \text{ incompressible}}$.

The altitude can be approximated by varying the RHO input to correspond to the altitude desired.

c. Time Estimating

Computer time for this program is a function of modes requested, time sharing, priority, and aerodynamics. Generally, for multiple problems within a run allow three minutes for the first case and two minutes for each additional case. Clock time on an IBM 360/65 runs about 2:1.

C. DFAL18 - FPR Blade Folding Response Analysis

Most of the inputs to DFAL18 are the same as used in DFAL17. A DFAL17 case must be run prior to running DFAL18 to generate the flexibility matrix and eigenvalues inputs to DFAL18.

The card order in a DFAL18 data deck is shown in Figure 11.

1. Input Format

CARD 01 Format 20A4:

Title Card; Column 1-80, alphameric.

CARD 02 Format I4:

Col.	1 - 4	NG	If NG is minus, then all response points will be printed. If NG = 1, sine wave gust. If NG = 2, sine-squared gust. If NG = 3, step gust. If NG = 4, pulse gust. If NG = 5, step gust with Wagner function.
Col.	5 - 8	NT	If NT = 0, calculate coordinates only. If NT = 1, calculate force, torque, and moments only. If NT = 2, calculate coordinates and forces, torques, and moments.
Col.	9 - 12	NP	Number of input flutter frequencies NP $\leq$ 10.
Col.	13 - 16	NSG	Number of coordinates to be run NSG $\leq$ 10.

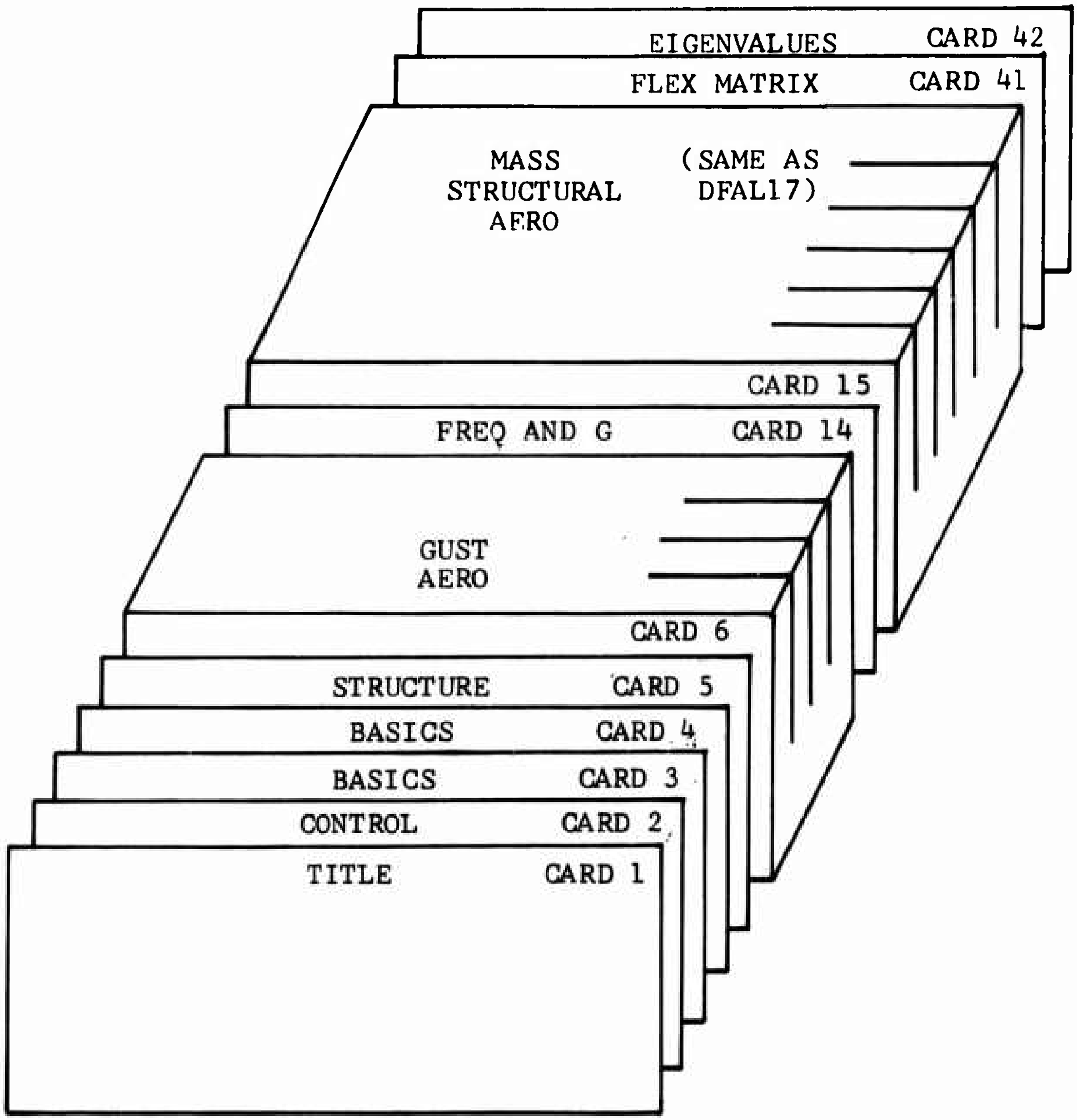


Figure 11. Card Order in DFAL18 Data Deck.

	Col. 17 - 20 21 - 27 ...	NSCB	Coordinate designation to be run, NSG values.	
CARD 03	Format E12.5:			
	Col. 1 - 12	RHO	Air density	(slug/ft^3)
	Col. 13 - 24	V	Aircraft forward velocity	(knots)
	Col. 25 - 36	CLA	Lift curve slope for gust	
	Col. 37 - 48	VG	Gust velocity	(ft/sec)
	Col. 49 - 60	BR	Reference semichord	(ft)
	Col. 61 - 72	S	Reference semispan	(ft)
CARD 04	Format E12.5:			
	Col. 1 - 12	DT	Incremental time step	(sec)
	Col. 13 - 24	OM	Blank, gust frequency or a pulse duration	(cps) (sec)
	Col. 25 - 36	TMAX	Maximum response time	(sec)
	Col. 37 - 48	FK	Flex matrix scale factor	
	Col. 49 - 60	WGSWEP	Wing sweep, minus (-) forward	(deg)
	Col. 61 - 72	BLDSWP	Blade sweep, plus (+) aft	(deg)

Card 05 is repeated for each coordinate to be analyzed, i.e. NSG cards are required.

CARD 05	Format E12.5:		
	Col. 1 - 12	EIB	(lb-in^2)
	Col. 13 - 24	GJ	(lb-in^2)
	Col. 25 - 36	L_1	(in)
	Col. 37 - 48	L_2	(in)
CARDS 6 - 9	Format 6E12.5: Col. 1 - 72	DY_i	(ft)
CARDS 10-13	Format 6E12.5: Col. 1 - 72	B_i	(ft)
CARD 14	Format E12.5:		
	Col. 1 - 12	G_1	
	Col. 13 - 24	ω_1	(cps)
	Col. 25 - 36	G_2	
	Col. 37 - 48	ω_2	(cps)
	Col. 49 - 60	G_3	
	Col. 61 - 72	ω_3	(cps)

Card 14 is repeated until NP damping and frequency pairs have been input--three to a card. A maximum of 10 frequencies is allowed.

Cards 15 through 30 are identical with Cards 4 through 19 of DFAL17. Cards 31 through 40 are identical with Cards 32 through 41 of DFAL17.

CARD 41	Format 6E12.5:	
	Col. 1 - 72	
CARD 42	Format 6E12.5:	
	Col. 1 - 72	

2. <u>Guide to Input Format</u>

CARD 01	TTL	is the title card consisting of alphabetic or numeric information required for identification of the run. This card is reproduced on the first sheet of output and on the plots.
CARD 02	NG	is a control number designating the gust shape. The gust begins at time = 0.

If NG > 0, a plot tape is generated.

If NG < 0, all response points and loads will be printed as well as plotted.

If NG = 1, a continuous sine wave with frequency defined by OM (Card 4) in cps and magnitude defined by VG in ft/sec.

If NG = 2, a half cycle, 1-cosine wave with frequency defined by OM in cps and magnitude defined by VG in ft/sec.

If NG = 3, a step gust input with magnitude defined by VG in ft/sec.

If NG = 4, a pulse gust with a duration defined by OM in seconds with a magnitude defined by VG in ft/sec.

If NG = 5, a step gust modified by the Wagner function. The Wagner function is approximated as:

$$1 - 0.165\ e^{-0.4555} - 0.335\ e^{-0.3S}$$

where $S = \frac{Vt}{B}$

V = forward velocity, ft/sec

t = time seconds

B = reference semichord, ft

The magnitude defined by VG in feet per second.

NT — is an option designating the output required. Consideration should be given to minimize the output since each additional output requires computer time.

If NT = 0, only the coordinate responses will be calculated.

If NT = 1, only the forces, moments, and torques will be calculated.

If NT = 2, the forces, moments, torques, and displacements of all coordinates will be calculated.

NP — is the number of coupled modes to be input, NP $\leq$ 10.

NSG — is the total number of coordinates to be considered, NSG $\leq$ 10.

NSCP — is the number of each coordinate to be run. There will be NSG coordinates entered successively in I4 format.

CARD 03

RHO — is the air density in slug/ft^3.

V — is the aircraft forward velocity (knots).

CLA — is the lift curve slope of the lifting surfaces, i.e. wing and three blades. This lift curve slope is used only for the gust induced airloads.

VG — is the maximum vertical gust velocity in feet per second.

	BR	is the reference semichord (ft). Should be the same as that used in the flutter analysis DFAL17.
	S	is the reference semispan (ft). Should be the same as that used in the flutter analysis DFAL17.
CARD 04	DT	is the incremental time for response in seconds. This increment should be small enough to allow twelve points for the smallest frequency of interest. Space is provided for up to a 5-second response at 0.0005 second increments. That is, 1000 time points can be run for each coordinate. There is no need to run a response case for long periods of response time since the points of interest usually are the starting transient and the resultant steady state. Both of these are easily achieved within 1000 time points.
	OM	depending on gust option (NG), selected OM could be a blank (step), gust frequency (sine wave, 1-cosine, cps) or a pulse duration (sec).
	TMAX	Maximum time for response in seconds. $\frac{TMAX}{DT} \leq 1000$
	FK	is the flexibility matrix scale factor which must be the same as FK used in DFAL17 used to scale the flexibility matrix.
	WGSWEP	is the wing elastic axis sweep angle in degrees. Forward sweep is negative.
	BLDSWP	is the blade elastic axis sweep angle in degrees. Folding aft is positive.

CARD 05 One card must be supplied for each coordinate that is to be analyzed. This card provides the structural parameters required to calculate internal loads at the coordinate.

EIB	is the beamwise EI for the segment corresponding to node being analyzed (Table I) in $lb\text{-}in^2$

	GJ	is the torsional stiffness for segment corresponding to node being analyzed, Table I.
	L_1	is the length from the origin to the left node of the segment being analyzed. Can be any length desired but must be ΔL shorter than L_2, $\Delta L = L_2 - L_1$. For the wing L_1, is the length from the root to node left. For the blade is the length from the intersection of the mast centerline and the blade elastic axis.
	L_2	is the length from the origin to the right node of the segment being analyzed. Can be any length desired, but must be ΔL longer than L_1, $\Delta L = L_2 - L_1$. For the wing, L_2 is the length from the root to the node right. For the blade, L_2 is the length from the intersection of the mast centerline and the blade elastic axis.
CARDS 6 - 9	DY	is the width (in feet) of the aerodynamic surfaces used to compute the gust loadings on the wing and blades. These are the DYs used in DFAL17, except that for blades 2 and 3, the DYs must be one-half of those for blade 1. This is because blades 2 and 3 have one-half of the gust velocity normal to their planform. These DYs should reflect span and aspect ratic effects. The DYs are entered in order on four succeeding cards in the order:
	CARD 06	Wing - five segment pylon plot
	CARD 07	Pylon yaw and blade 1 all five segments
	CARD 08	Blade 2 - five segments, Blade 3 - one segment
	CARD 09	Blade 3 - segments two to five.
		DY can be zero.
CARDS 10-13	B	is the local semichord (in feet) of the aerodynamic surfaces. B is used to compute the gust loads on the wing and blades. These Bs would normally correspond to the Bs used in DFAL17. The Bs are entered in order on four succeeding cards in the order:

	CARD 10	Wing (five segments) and pylon pitch
	CARD 11	Pylon yaw and blade 1 (five segments)
	CARD 12	Blade 2 (five segments), and first segment of blade 3
	CARD 13	Blade 3 (segments two through five).
		B cannot be zero.
CARD 14	NP	G_i and ω_i are the damping factor and frequency in cps, respectively. These are obtained from the DFAL17 output for the velocity input to DFAL18. NP pairs must be input.

Cards 15 through 30 are identical with Cards 4 through 19 of DFAL17.

Cards 31 through 40 are identical with Cards 32 through 41 of DFAL17.

CARD 41	FLEX	is the structural influences coefficient matrix generated with DFAL17. This matrix is 74 x 74 and consists of 962 cards, sequentially numbered and in 6E12.5 format. If a card is out of order, the program will stop and print an error message identifying the last good card.
CARD 42	CMPLXR	are the complex eigenvalues for the first reduced velocity and are generated using DFAL17. The reduced frequency should be zero. CMPLXR consists of 25 cards, sequentially numbered.

3. <u>Output Format and Guide</u>

DFAL18 output of DFAL18 consists of four parts:

- All input data.
- The calculated aerodynamic matrix and eigenvalues.
- A printed time history of the coordinate displacements, forces, moments, and torques.
- A machine (CALCOMP) plot of displacements and the forces, moments, and torques at the right and left nodes.

The output format is shown by a sample case in Appendix I.

4. Suggested User Techniques

a. Problems Encountered

As in Program DFAL17, there are peculiarities associated with the program. Given below are the most frequent complaints.

(1) Loads calculated are too high or too low. This can be caused by any of the following:

(a) Input data error; card out of place, integer number not right oriented or floating number out of field.

(b) Incorrect number of modes. Since blade modes occur in threes, to include one or two will result in erroneous loads.

(c) GJ or EI for the segment is wrong.

(d) Loads are correct and the users estimater is off.

(2) The job will terminate if a card in the flexibility matrix is out of order. All cards punched as output from DFAL17 are sequentially numbered. This sequence number is checked in DFAL18.

b. Time Estimate

The approximate run time for the loads at one node is approximately six minutes. This time will vary with priority time sharing procedures and whether output is printed on line or off line.

SECTION III

PROGRAMMING GUIDES

This section contains the fundamental structure of programs developed under this contract. Details of numerical methods and program organization are then discussed. FORTRAN listings will not be included in this section; they are listed in Appendix II.

A. ARAP06

1. Method of Digital Computer Solution

A predictor-corrector integration technique is used in the solution of the simultaneous equations of motion which are nonlinear, open-form in nature. In general the solution of a set of differential equations by this method requires the isolation of the highest order derivatives. For example given the equation

$$a\ddot{x} + b\ddot{x} + cx = F(t)$$

where the coefficients a, b, c and forcing function, F(t), may all be time dependent and/or nonlinear.

To solve for $\ddot{x}$, the equation is rearranged in the following form

$$x = \frac{F(t)}{a} - \frac{b}{a}\dot{x} - \frac{c}{a}x$$

Prior to evaluating this equation, a value of $\ddot{x}$, called $\ddot{x}_g$, is guessed. When an iterative procedure starts at a new time point, $\ddot{x}_g$ is used from the solution at the previous time point. Otherwise it is predicted by use of point-slope computational methods to minimize the difference between the value calculated and the value guessed.

Since values of velocities and displacements are also required, to evaluate the equation Taylor's series is used to provide a numerical integration form for determining them as functions of the guessed acceleration.

By definition the integral of a function f'(x) over the interval h is:

$$f(x+h) - f(x) = \int_{x}^{x+h} f'(x)dx$$

Also the incremental form of Taylor's series expansion of a fraction f(x) is:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{6}f'''(x) + \ldots$$

Thus the integral

$$\int_x^{x+h} f'(x)dx = hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \ldots$$

Defining derivatives of f(x) in the following way:

$$f'(x) = \frac{f(x) - f(x-h)}{h}$$

We can produce an integration form which permits evaluation of f(x+h) in terms of its derivatives, i.e. numerical integration. The derivative can be defined in other ways, for example

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

But for simplicity of programming the first definition is used here.

Substituting and letting $h = \Delta t$:

$$f(x+\Delta t) = f(x) + \Delta t \cdot f'(x) + \frac{(\Delta t)}{2}\left|f'(x+\Delta t) - f'(x)\right|$$

$$+ \frac{(\Delta t)^2}{6}\left|f''(x+\Delta t) - f''(x)\right| + \ldots$$

Thus

$$f(x+\Delta t) = f(x) + \frac{\Delta t}{2}\left|f'(x+\Delta t) + f'(x)\right|$$

$$+ \frac{(\Delta t)^2}{6}\left|f''(x+\Delta t) - f''(x)\right| + \ldots$$

Let

$$f(x+\Delta t) = \dot{x}_g$$

retain the second term only, one obtains,

$$\dot{x}_g = \dot{x}_{n-1} + \left(\ddot{x}_g + \ddot{x}_{n-1}\right)\frac{\Delta t}{2}$$

Then let

$$f(x+\Delta t) = x_g$$

one obtains

$$x_g = x_{n-1} + \left(\dot{x}_g + \dot{x}_{n-1}\right)\frac{\Delta t}{2} + \left(\ddot{x}_g - \ddot{x}_{n-1}\right)\frac{(\Delta t)^2}{2}$$

where the subscript n-1 refers to the previous time point.

A computed acceleration is then obtained by using the guessed values of $\dot{x}_g$ and x_g, i.e.

$$\ddot{x}_c = \frac{F(t)}{a} - \frac{b}{a}\dot{x}_g - \frac{c}{a}x_g$$

This computed value in general will not be equal to the guessed one. A convergence criterion has to be met such that

$$\frac{\left|\ddot{x}_c - \ddot{x}_g\right|}{\left|\ddot{x}_c\right|} \leq \varepsilon$$

where ε is the error tolerance.

If the error percentage fails to be less than ε, a quadratic point-slope subroutine is called using parabolic curve fitting to minimize the error percentage to zero. A better predicted $\ddot{x}_g$ is then obtained for the next iteration. The process repeats itself until the convergence criterion is met. Throughout these prediction and correction procedures, the independent variable-time remains constant. When the error percentage is within the tolerance, the convergence is completed. The value of time is then increased by an increment and the iterative process is repeated.

2. <u>Subroutine Descriptions</u>

a. <u>FORTRAN Subroutines</u>

The FORTRAN subroutines contained in the Proprotor Aeroelastic Analysis, ARAP06, are listed in alphabetical order except for the main program which is first. A few remarks are made for each subroutine, which indicate the general purpose or use of the subroutine in the general structure of the program. Great detail of explanation is not attempted here since it would be redundant.

(1) MAIN - This routine is the main program of ARAP06. This subprogram sets up the time loop and directs the flow of beginning and ending of the whole program. The rate of feathering or unfeathering of rotor-blades is computed here.

(2) AIRLDS - This subroutine handles the computation of rotor aerodynamics.

(3) BEXIOM - Tabulated forms of natural frequencies, normal modes and bending moments of flexible blades as functions of rotor rpm and blade feathering are read in. Printing out of these tables is also included in this subroutine.

(4) BLOCK DATA - This subroutine contains tabulated forms of blade aerodynamic lift, drag, and moment coefficients C_L, C_D, and C_M as functions of Mach number and angle-of-attack. It also contains the tabulated coefficients of blade pitch-damping as functions of angle-of-attack and reduced velocity.

(5) CLCDR - A maximum of four different blade airfoil sections can be handled in ARAP06. This subroutine determines the airfoil for each blade segment. This subroutine also provides the interpolation for values of C_L, C_D, and C_M for the most inboard blade airfoil section.

(6) FLAPN1 - Forcing functions for blade rigid-body flappings and rotor rotational degrees-of-freedom are calculated. These equations are then solved simultaneously.

(7) FLAPN3 - Tests of converging criteria for blade rigid-body flappings and rotor rpm are conducted. If convergence is completed, forces and moments due to rotor aerodynamics, rotor inertials and rotor gravity are summed and ready to be used as a forcing function for wing/pylon dynamic equations. Print-out of converged rotor variables is then processed. The peaks of master blade flapping are stored if required.

(8) FLEXBD - This subroutine solves for generalized coordinates for the flexible blades.

(9) FLEX1 - Because of limitations on the size of subprograms in the FORTRAN compiler, all dynamic terms used in the dynamic equations of motion for rotor-blades cannot be handled in any single subroutine. Some terms are, therefore, calculated in this subroutine and will be used in both subprograms FLAPN1 and FLEXBD.

(10) FORCES - Rotor inertial loads and moments are calculated when the convergence of blade generalized coordinates is completed.

(11) HEADNG - This subroutine handles the FORMAT of the print-out headings. The input options have control on the selection of variables to be computed as well as on the output headings.

(12) INPUT - The read-and-write of input data is done in subroutines BEXIOM and READWR. The subroutine INPUT changes input parameter units if necessary, and computes time-invariant quantities from input data.

(13) PLTS - This subroutine plots the specified variables.

(14) PSLOP - This subroutine minimizes the difference between the calculated acceleration and the predicted one and provides a better prediction for next iteration. Let $\ddot{x}_g$ be the predicted acceleration, x_c be what calculated, and $\Delta\ddot{x} = \ddot{x}_c - \ddot{x}_g$ a point, $(\ddot{x}_g, \Delta\ddot{x})$ is then obtained in a two-dimensional space. When iteration loop starts at a new time point, there is only one $(\ddot{x}_g, \Delta\ddot{x})$ available. The next prediction is given by simply averaging $\ddot{x}_g$ and $\ddot{x}_c$. After the second iteration, there are two points of $(\ddot{x}_g, \Delta\ddot{x})$ available. A straight line passing through these two points and intersecting the axis of abscissa gives a value of $\ddot{x}_g$. This $\ddot{x}_g$ is used as a predicted value for the next iteration. After the third iteration, there should always be three points of $(\ddot{x}_g, \Delta\ddot{x})$ available, a quadratic equation is then used. The intersection of the curve represented by this quadratic equation and the abscissa provides the next predicted $\ddot{x}_g$ for use in the following iteration. Two subroutines are generally obtained by solving a quadratic equation, the one which is closest to the last guess is used.

(15) READWR - This subroutine reads and writes all the input data for proprotor and wing/pylon except those for the rotor flexible blades.

(16) RSTN - Flapping restraint and stops of a gimballed rotor are handled in this subroutine.

(17) TABINA - Interpolation of tabulated forms of natural frequencies, normal modes and bending moments of flexible blades as functions of instantaneous rotor rpm and blade feathering is done in this subroutine.

(18) TABINT - Excluding the most inboard blade airfoil section, the C_L, C_D, and C_M of the rest of the three sections of a blade are provided by interpolation.

(19) WING - Wing aerodynamics and generalized force for wing/pylon system are calculated. Dynamic equations of motion of wing/pylon are then solved. Converging criteria for the solved normal coordinates are tested. Pylon accelerations after completion of the convergence are computed.

b. Subroutines Cross-Reference

The information provided in Table I is vital for understanding of the subroutine organization of program ARAP06. The first column contains the names of the subroutines in alphabetical order, except for the main program which is first. The second column, length, contains the size, in bytes in hexadecimal number base, of the compiled subprogram. The FORTRAN subroutines, all but BLOCK DATA, have been compiled on the IBM operation system 360 FORTRAN IV (H) Compiler with OPT = 2. The subroutine, BLOCK DATA, has to be compiled on FORTRAN IV (G) Compiler because of its size. The third column contains two operations, and the fourth column contains the names of the subroutines associated with that operation and given in the order of call sequence.

3. Flow Outline

The flow charts in Figures 13 and 14 describe the functional structure of the program and the logic used in the subroutine AIRLDS respectively, without regard to flow by subroutines. Figure 13 provides a clear picture of the numerical technique used in ARAP06 to solve the equations-of-motion simultaneously. Figure 14 demonstrates how the wing-rotor aerodynamic interference is handled in the calculation of blade airloads. Tedious flow charts are not given in this report. The reader can refer to FORTRAN listings for details.

TABLE I

ARAP06 SUBROUTINE CROSS-REFERENCE

Name	Length	Operation	Cross-Reference
MAIN	CE0	Calls	INPUT, HEADNG, FLEX1, AIRLDS, FLEXBD, FLAPN1, FLAPN3, PLTS
AIRLDS	147A	Calls	CLCDR
		Is Called By	MAIN
BEXIOM	1274	Is Called By	READWR
BLOCK DATA	2A08	Is Used By	TABINT
CLCDR	930	Calls	TABINT
		Is Called By	AIRLDS
FLAPN1	1274	Calls	RSTN
		Is Called By	MAIN
FLAPN3	1814	Calls	PSLOP, TABINA, FORCES, RSTN, WING, HEADNG
		Is Called By	MAIN
FLEXBD	1ECA	Calls	PSLOP
		Is Called By	MAIN
FLEX1	94C	.Is Called By	MAIN
FORCES	F1E	Is Called By	FLAPN3
HEADNG	56C	Is Called By	MAIN, FLAPN3
INPUT	13A6	Calls	READWR, TABINA
		Is Called By	MAIN
PLTS	2A2C	Is Called By	MAIN
PSLOP	41A	Is Called By	FLAPN3, FLEXBD, WING
READWR	24E6	Calls	BEXIOM
		Is Called By	INPUT
RSTN	304	Is Called By	FLAPN1, FLAPN3
TABINA	5F0	Is Called By	INPUT, FLAPN3
TABINT	9C2	Is Called By	CLCDR
WING	12C0	Calls	PSLOP
		Is Called By	FLAPN3

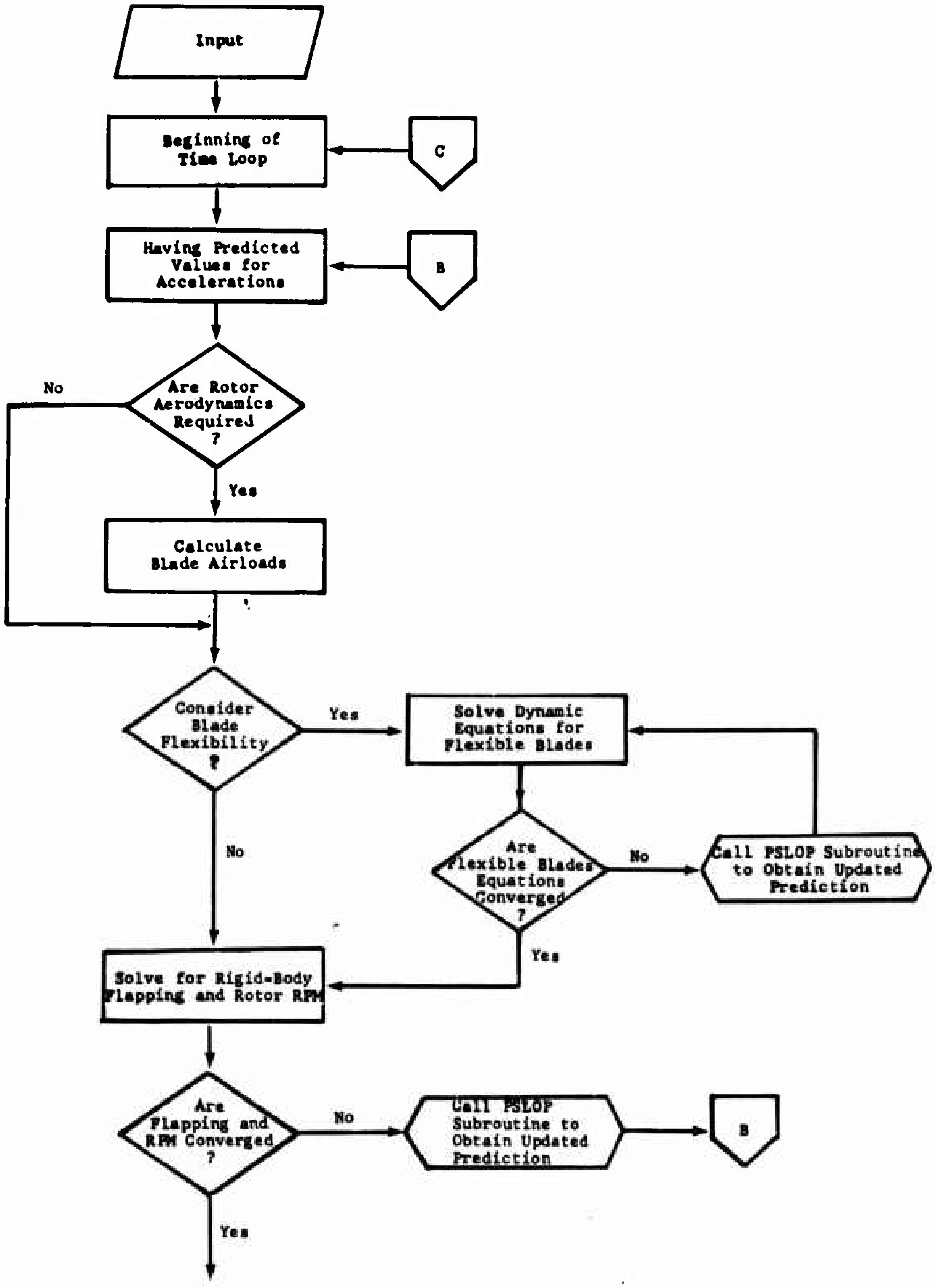


Figure 12. Flow Chart for Program Structure - ARAP06.

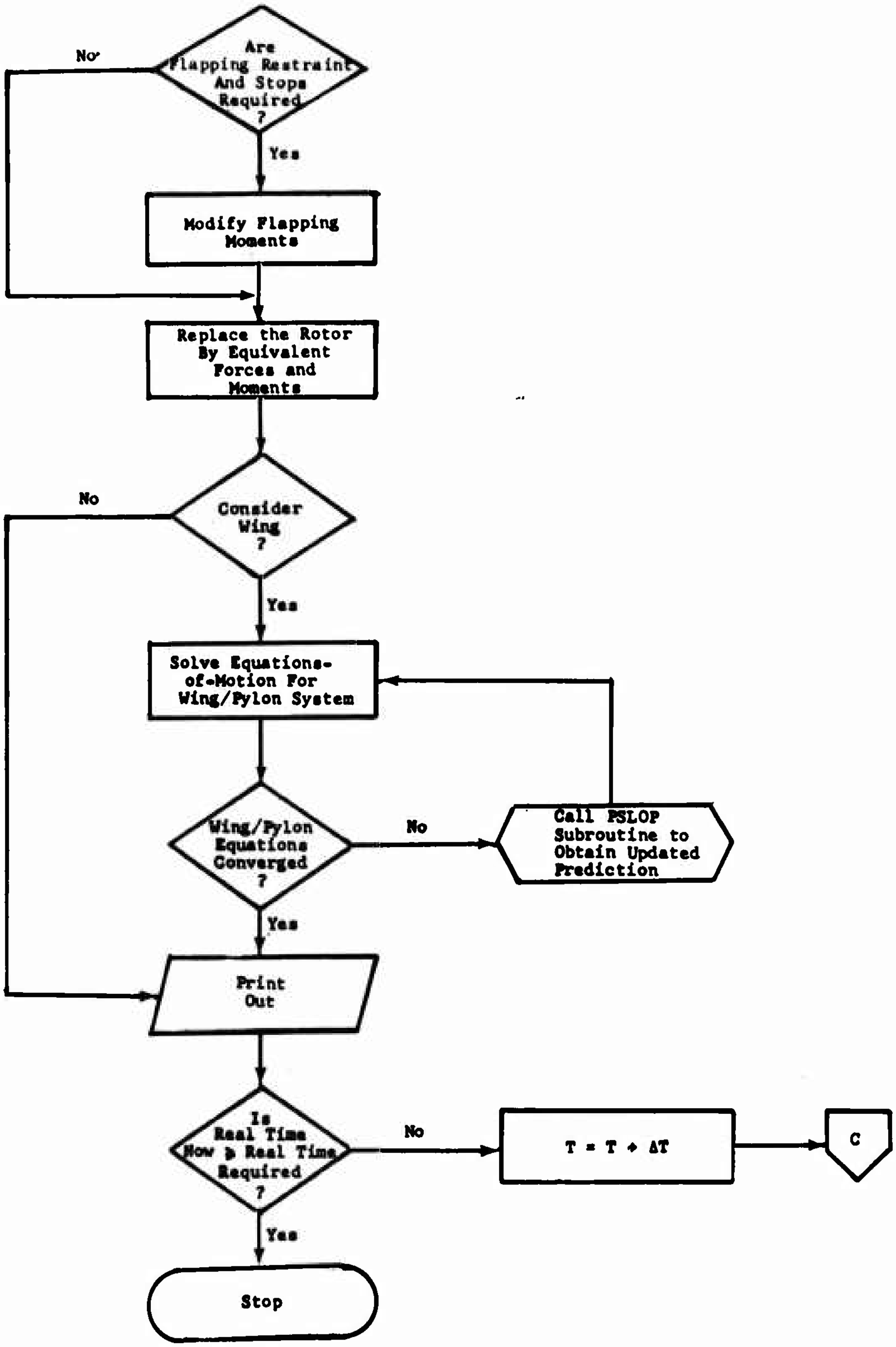


Figure 12. Concluded

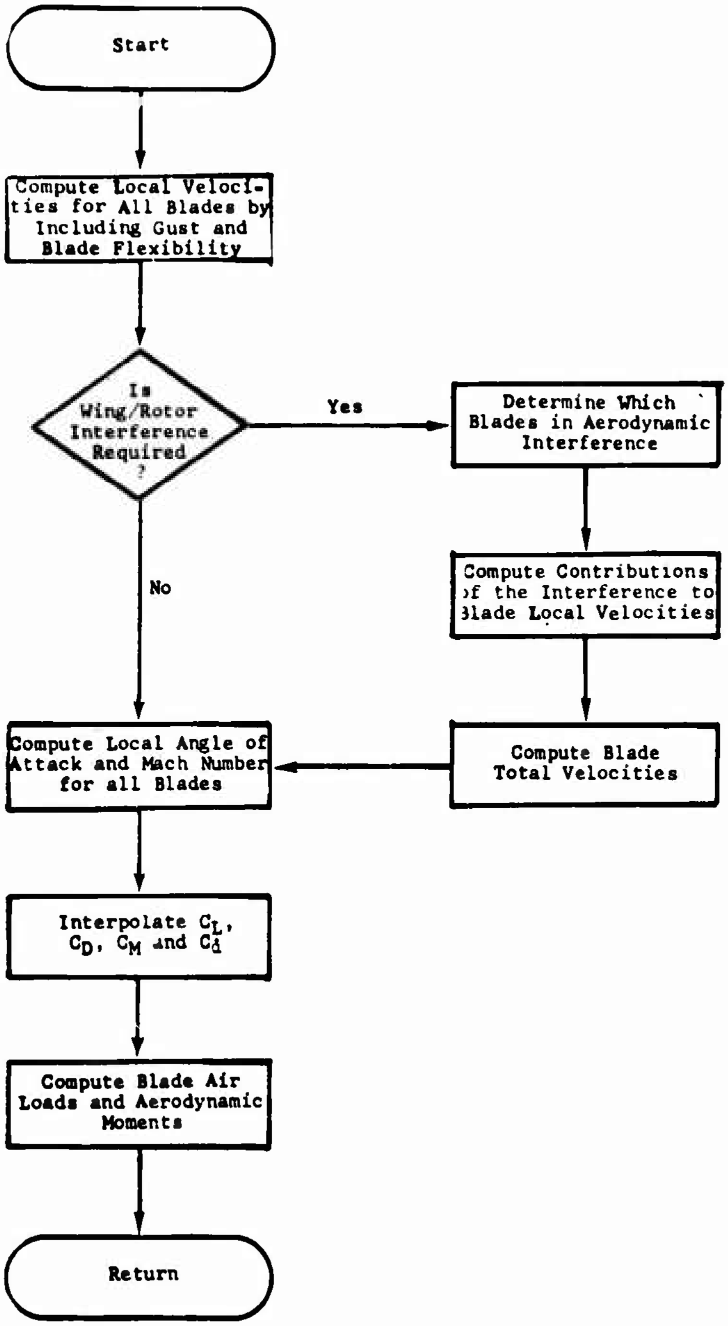


Figure 13. Flow Chart for Subroutine AIRLDS.

B. DFAL17

1. Method of Solution

Computer program DFAL17 solves for the flutter speed of a folding proprotor. The stiffness matrix, mass matrix, and aerodynamic matrices are constructed, arranged and solved with a minimum of input data. The user needs only to specify stiffnesses, geometric sizings, and fold angles.

The dynamics matrix is solved by ALLMAT, a QR algorithum, which solves for all 74 eigenvalues in a minimum of computer time. The eigenvalues are then arranged in order of decreasing real part (increasing frequency) and the appropriate number of eigenvectors are calculated. The structural influence coefficient matrix and 74 eigenvalues can be punched output for use as input to computer program DFAL18.

The flow of the computer program is shown by a macroscopic flow diagram (Figure 13). A brief description of each subroutine, a flow outline, and a list of variable arrays are given on the following pages. The main variable arrays are defined at the end of this chapter. The reader is referred to the FORTRAN listing (Volume III) for program details.

2. Subroutine Description

SUBROUTINE PRT3

Initializes and builds the dynamics matrix using an assumed reduced velocity or flutter frequency.

SUBROUTINE PRT4

Solves the dynamic system for all eigenvalues and requested eigenvectors. Orders eigenvalues for lowest frequency, outputs eigenvalues, eigenvectors, natural frequencies, and punched eigenvalues if necessary.

SUBROUTINE DFAE, BESJ, BESY

This subroutine and its associated routines BESJ, BESY, form the aerodynamic influence coefficient matrix corresponding to a given reduced velocity or flutter frequency. BESJ computes J Bessel functions. Methods used are those given in IBM SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE, VERSION III PROGRAMMER'S MANUAL, H20-0205-3, page 363-364.

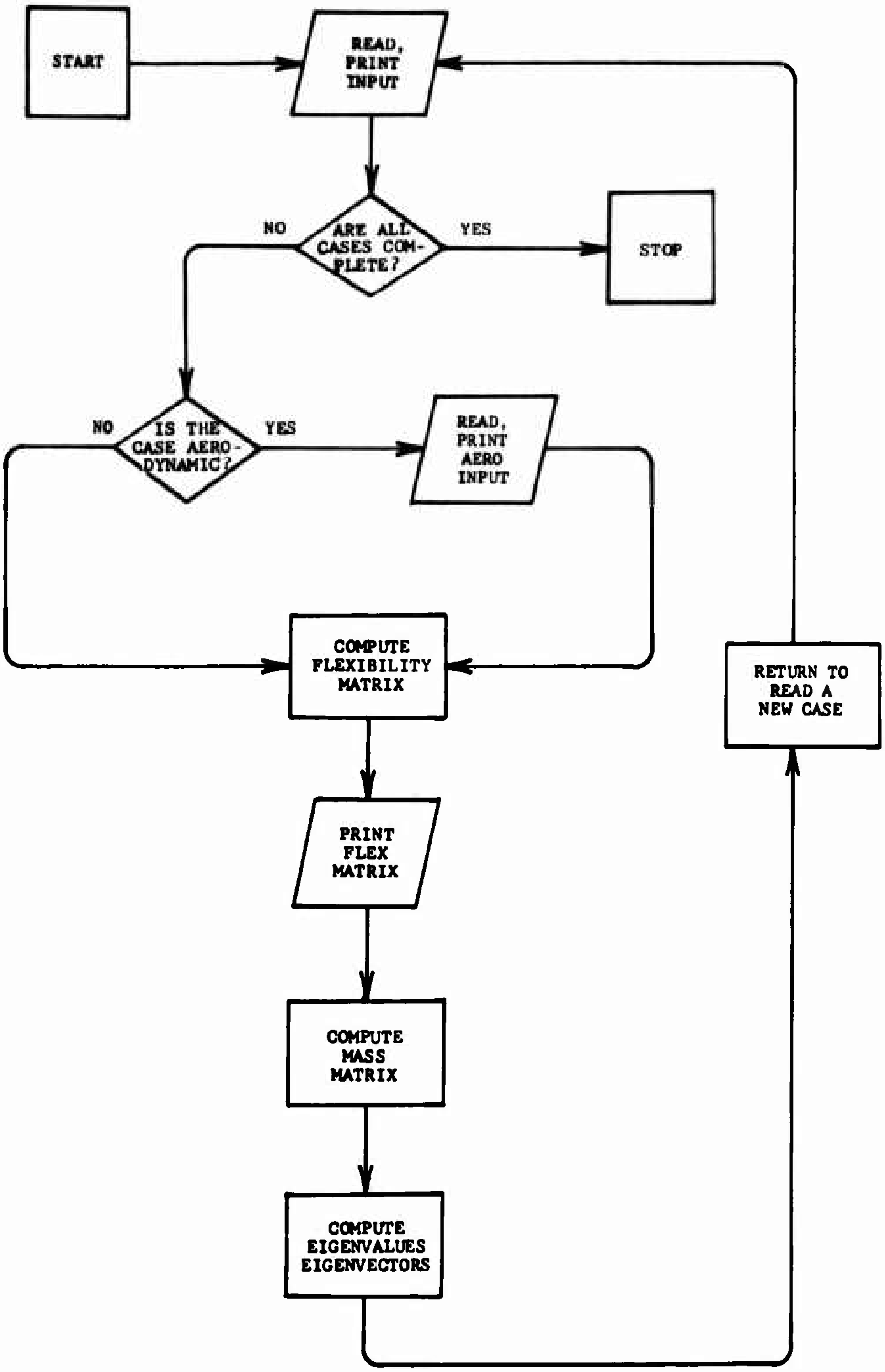


Figure 14. DFAL17 Flow Diagram.

SUBROUTINE STGN

Reads stiffness inputs, forms stiffness matrix parameters, inverts, transforms, and produces flexibility matrix. Punches flex matrix if necessary.

SUBROUTINE ROTATE

Builds large stiffness matrix, 180th order, and transformation matrix.

SUBROUTINE MPRINT

Prints a program matrix in column form.

SUBROUTINE REDUCE

Eliminates undesired coordinates from the symmetric stiffness matrix by Gaussian pivotal reduction, reduction of degrees of freedom from 180 to 74.

SUBROUTINE MULT

Forms a matrix product UA where A is a real matrix, in this program a flex matrix, and U is a complex aerodynamic-inertial matrix.

SUBROUTINE ALLMAT

This program uses a QR algorithm with special attention to minimization of required storage to solve for the eigenvalues and eigenvectors of a general system of the form $A - \lambda I = 0$, where A is a complex valued matrix. In this case the matrix A is 74th order, and not symmetric.

SUBROUTINE INVERS

This routine inverts an N x N matrix by Gaussian row and column pivot operations.

SUBROUTINE TPROD

This subroutine performs a specific matrix triple product X^TAX where A is a 180th order symmetric stiffness matrix in upper triangular form and X is a quinary diagonal transformation matrix.

SUBROUTINE DPROD

Performs a matrix triple product X^TAX where A is a symmetric N x N matrix and X is a general transformation matrix.

SUBROUTINE ERRSET

This IBM 360 System subroutine suppresses underflow error print ›its which occur from solution of an undamped dynamics matric problem with an eigenvalue routine, ALLMAT which is intended for solution of damped systems. In some cases, the imaginary part of the eigenvalues approaches zero slowly.

3. Flow Outline

MAIN

(1) Begin reading input for new case. Define default parameters. If no aero input is to be called for, to to (3).

(2) Read aero input and call subroutine DFAE to set up aero coefficients.

(3) Call subroutine STGN to compute flexibility matrix.

(4) Output flexibility matrix and scale by k factor.

(5) Start loop to evaluate distributed mass matrix.

(6) Form mass matrix by triple product coordinate transformation. When all mass elements are complete, go to (7).

(7) Print mass matrix and call subroutine PRT3 for eigenvalue-eigenvector computations.

SUBROUTINE PRT3

(1) Set up loop to build and evaluate system for each reduced velocity input in MAIN (2) or once for the structural system if no aero exists.

(2) Zero and build aero matrix transpose by calling subroutine DF67.

(3) Add the mass matrix transpose to the aero matrix. Units of slugs are consistent.

(4) Form product $(M+A)K^{-1}$, the augmented mass matrix post-multiplied by the flexibility matrix.

(5) Transpose the dynamics matrix transpose to form the dynamics matrix.

(6) Call subroutine PRT4 to compute eigenvalues and eigenvectors of dynamics matrix.

(7) If all reduced velocities have been run return to MAIN, otherwise go back to (2) for next velocity.

SUBROUTINE PRT4

(1) Compute all 74 eigenvalues by calling subroutine ALLMAT.

(2) Use loop to order the first MODES eigenvalues in ascending order of magnitude of the real or frequency component only. This corresponds to lowest flutter frequencies first.

(3) Print eigenvalues. If punch option is on, punch eigenvalues.

(4) Start loop to evaluate eigenvectors.

(5) Call subroutine entry ALLVEC to compute the eigenvector corresponding to a given eigenvalue.

(6) Normalize the eigenvector by division by the largest element.

(7) Go through logic to print eigenvectors in groups of threes.

(8) If all eigenvectors needed have been computed, go to (9), otherwise to (4) for next eigenvector.

(9) Start loop to evaluate all frequency and damping factors derived from flutter frequencies of interest.

(10) Compute and print flutter frequency, damping and equivalent velocity.

(11) At end of loop (9) return to subroutine PRT3.

SUBROUTINE DFAE

(1) Read input aero data case and print all quantities input. Return to MAIN program.

(2) At entry DF67 the reduced velocity EKR to be used is input to the subprogram.

(3) Start loop over all surfaces, in this program, four.

(4) Start inner loop over all strips on each surface.

(5) Set up transformation matrix A dependent on surface-strip geometry.

(6) Compute J and Y Bessel functions for the modified frequency.

(7) Set up the oscillatory aerodynamic coefficient matrix B using Bessel functions from (6).

(8) Do a transformation matrix triple product $A^T BA$ to get aerodynamic influence coefficient matrix.

(9) Scale the coefficients by an eiquivalent air mass factor.

(10) Use a loop to set up the transpose of the aerodynamic matrix by distributing the aerodynamic influence coefficients along the diagonal.

(11) Have all strips been computed? If not return to (4).

(12) Have all surfaces been computed? If not return to (3).

(13) Return to subroutine PRT3 with complex aerodynamic matrix.

SUBROUTINE STGN

(1) Read structural inputs and print.

(2) Use loop to initialize angles and lengths.

(3) Set up quantities required for stiffness matrix.

(4) Call subroutine ROTATE to build 180th order stiffness matrix A and transformation matrix B.

(5) Call subroutine TPROD to form triple product $B^T AB$.

(6) Call subroutine REDUCE to reduce large stiffness matrix to 74th order by elimination of coordinates.

(7) Read parameters required for mass matrix and coordinate transformation.

(8) Interchange columns 11, 12, and 13 in stiffness matrix and initialize transformation matrix.

(9) Invert stiffness matrix to get 74th order flexibility matrix.

(10) Print computed stiffnesses and input from (7).

(11) Build transformation matrix from input of (7).

(12) Call subroutine DPROD to do transformation triple product.

(13) If punch option is on, punch the final flexibility matrix.

(14) Return to MAIN.

SUBROUTINE ROTATE

(1) Zero upper triangular stiffness matrix storage.

(2) Start loop to build 180th order upper triangular stiffness matrix.

(3) Look at each node point, 30 in all; build the elements of the stiffness matrix by coordinate transformations of a coupled 2 node stiffness system.

(4) Store all computed values of node stiffness quantities in an upper triangular form due to symmetry.

(5) If all nodes have been completed go to (6), otherwise to (2).

(6) Initialize node independent values of the gamma transformation matrix.

(7) Build the gamma transformation matrix as a set of 3 x 3 matrices, 2 for each node.

(8) When the entire gamma transformation matrix is complete, return to subroutine STGN.

SUBROUTINE REDUCE

(1) Start loop to evaluate all 180 degrees of freedom.

(2) If the selected degree of freedom is not to be reduced to (8).

(3) Select pivot element of coordinate to be reduced.

(4) Use loop to build a row vector for use in reduction.

(5) Start loop to modify the row vector by division by the pivot element.

(6) Use inner loop to subtract the divided row from all other rows.

(7) When all row elements defined in (4) have been modified as in (5) go to (9), otherwise back to (5).

(8) Increment counter of the coordinates to be retained.

(9) If all coordinates have been analyzed, go to (10), otherwise back to (1) for new coordinate.

(10) Use loop to do reduction of triangular 180th order elements to be kept, to a triangular 74th order form.

(11) Transfer 74th order symmetric form to a 74th order square matrix. Reason for the double step ((10) - (11)) is because of space limitations.

(12) Return to subroutine STGN.

4. Variable Arrays

(* Signifies Input Parameter)

* VELCTY(15)	Values of reduced velocities
A(74,74)	Flexibility matrix
C(2,2), B(2,2)	Transformation matrices
* X(22) * Y(22) * M(22) * SX(22) * IX(22)	Mass matrix parameters for building distributed mass matrix and also for forming dumbbell transformation matrix
AM(150)	Mass matrix element values. Only nonzero values are saved.
ICT(150)	Mass matrix subscript value, respective storage position stored columnwise.
U(74,74)	Dynamics matrix work area
VS(74,3)	Storage for up to three eigenvectors for printout purposes
EIG(74)	Eigenvalues computed
VECT(74)	Work area for eigenvector computation
NCB(3)	Numbers of eigenvectors computed
FMT(3,3)	Storage for variable eigenvector formats

FMA(3)	Work area for eigenvector formats
* COSLMA(4) * BRA(4) * SA(4) * CAPA(4)	Storage for surface variables input to aero coefficient evaluation
* DELTAY(7,4) * B(7,4) * D(7,4)	Storage for surface and strip variables input to aero coefficient evaluation
AM(2,2), BM(2,4) CH1(2,4), PARTA(2,4)	Work areas for use in aero coefficient evaluation
* XDL(5,2) * XDEIB(5,2) * XDEIC(5,2) * XDGJ(5,2)	Lengths, bending stiffnesses, torsional rigidity for all coordinates, for building stiffness matrix
ALPHA(30),PSI(30) BETA(30),EIB(30) EIC(30),GJ(30), AE(30), L(30)	Transformation angles and various other quantities used to set up original stiffness matrix
IX(74)	Subscript list of variables to be kept during reduction of 180th order stiffness matrix to 74th order
STIF(74,74)	Reduced stiffness matrix
* PARM(22,5)	Same as X,Y,M,SX,IX above
FINAL(74,74)	Transformation matrix for flexibility modification
REDK(16290)	Upper triangular 180th order stiffness matrix
KMAT(12,12),AMAT(12,12) S(12)	Work areas for building large stiffness matrix
RED(180)	Column work area for reduction of upper triangular stiffness matrix
H92921)	Work area to hold upper Hessenberg form of the dynamics matrix during eigenvalue analysis
MULT(74),SHIFT(3), INTH(74),NS(74),ING(74)	Work areas for eigenvalue-eigenvector analysis

NSZ(74)	Transformation matrix for 180th order triple product
XTD(3,3),XTE(3,3)	Work areas for above triple product

C. DFAL18

1. Method of Solution

DFAL18 solves for the internal forces, moments, and torques acting at a specific node on a FPR wing, pylon, blade system, due to an externally applied airload. The solution involves determining the response of selected coordinates $q_n(t)$ and using the responses to compute internal loads.

The response of each coordinate is calculated by adding the time dependent transfer functions for each flexible mode. Each mode has the unsteady aerodynamics associated with its frequency. The transfer functions are expressed as polynomials in terms of the Laplacian operator s. While the inverse Laplacian could be formed a finite differencing integration technique is used to calculate the coordinate time responses. Internal loads are then calculated by multiplying the coordinate displacement times the elemental stiffness matrix. This product is done at each time step.

Coordinate responses and internal loads are shown as a function of terms on CALCOMP plots and, if desired, printed output.

The flow of the computer program is shown by a macroscopic flow diagram (Figure 15). A brief description of each subroutine with a written flow explanation is given on the following pages. The main variable arrays are defined at the end of this chapter. The reader is referred to the FORTRAN listing (Volume III) for program details.

2. Subroutine Descriptions

SUBROUTINE TMF

This subroutine sets the plotting axes, scales the variables, and plots the time gust function.

ENTRY HIST

This subroutine entry point prints and plots each force, moment, or coordinate displacement.

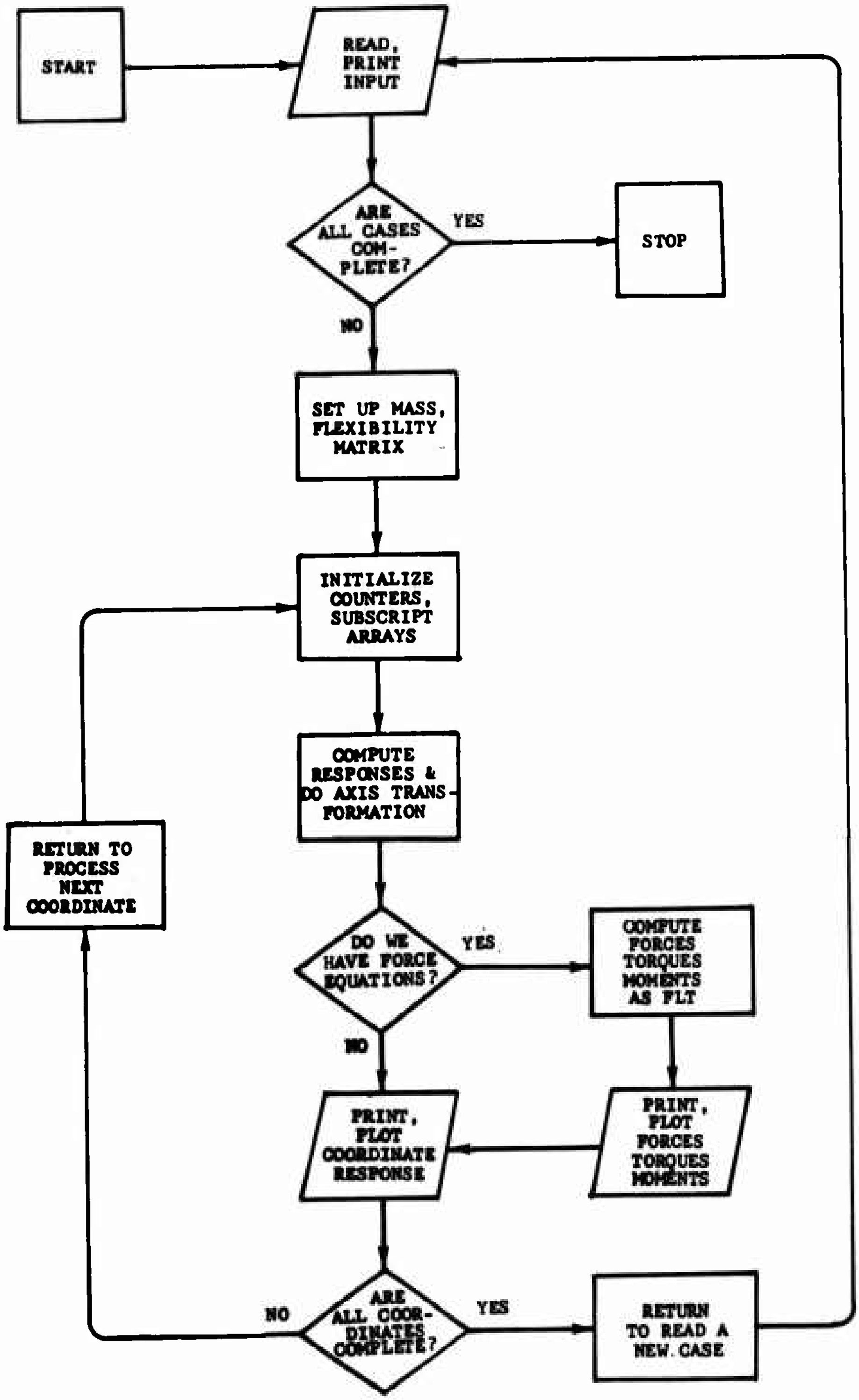


Figure 15. DFAL18 Flow Diagram.

SUBROUTINE RESP

This routine evaluates the time response of all second order transfer functions and accumulates the real part of the response for each transfer function to form a total forced response for each coordinate.

SUBROUTINE MASS

Forms a mass matrix corresponding to the mass matrix distribution of the generalized coordinates.

SUBROUTINE FN

This subroutine computes the time history of the gust input function. The routine may be simply modified or extended to provide any gust input desired, according to a user's needs.

SUBROUTINE INVDET

This subroutine computes the complex determinant value of a complex matrix by Gaussian pivotal reduction.

SUBROUTINE MULT

This subroutine forms a matrix product UA where A is a real matrix, in this program a flex matrix, and U is a complex aerodynamic-inertial matrix.

SUBROUTINE DPROD

This subroutine forms a matrix triple product A^TBA where B is a symmetric matrix, that is, $B = B^T$.

SUBROUTINE DFAE, BESJ, BESY

This subroutine and its associated routines BESJ, BESY, form the aerodynamic influence coefficient matrix corresponding to a given reduced velocity or flutter frequency. BESJ computes J Bessel functions of order 0 and 1, BESY does likewise for Y Bessel functions. Methods used are those given in IBM SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE, VERSION III PROGRAMMER'S MANUAL, H20-205-3, page 363-364.

SUBROUTINE RESCOM

This subroutine sets up, evaluates, and reduces each dynamics matrix formed. It then evaluates transfer function coefficients and time responses for all coordinates.

SUBROUTINE REDUCE

This subroutine eliminates undesired coordinates by Gaussian reduction, and retains the modified matrix and forcing vector for determinant evaluation.

SUBROUTINE ERRSET

An IBM 360 subroutine which suppressed underflow error printouts when the J Bessel functions are evaluated very near a zero value.

3. Flow Outline

(1) Initialize CALCOMP plot buffer storage for plot output.

(2) Start reading input for a new case. The termination point for a run when all cases have been completed is (23). Read all general input and control parameters and print.

(3) Compute constants used in iteration loops.

(4) Read wing parameters and compute gust force magnitude for each coordinate station.

(5) Read damping and frequency values obtained from DFAL17 for each reduced velocity and print.

(6) Call subroutine DFAE to initialize, read and print aerodynamic input parameters.

(7) Call subroutine MASS to read mass matrix inputs and compute mass matrix.

(8) Read flexibility matrix and flutter frequencies output from DFAL17.

(9) Compute flutter frequencies in λ form from input frequency and damping.

(10) Print flutter system roots and flexibility matrix.

(11) Convert flex matrix to foot units and form modified forcing vector F'.

(12) Start iteration loop on all coordinates at which a force and bending moment or a coordinate response is required.

(13) Initialize counters to indicate flutter frequencies and coordinate stations to be used and call subroutine RESCOM.

(14) Begin loop over all time points. This loop transforms all coordinate responses from the generalized form of two vertical displacements to the coordinates representing the dumbbell model, that is, one rotational and one vertical translational mode at each station.

(15) If only the translational-rotational coordinates are to be printed and plotted (input option NT), go to (21).

(16) Compute parameters required to compute forces, torques, moments.

(17) Start iteration loop to evaluate all quantities which are functions of the translational-rotational coordinate responses computed in (14).

(18) Compute bending slopes as functions of translations and torsional angles as functions of rotations and translations. If response at the first station is to be evaluated, set cantilever boundary condition.

(19) Compute forces, torques, moments.

(20) Call subroutine HIST to print and plot time histories of forces and moments computed in (19).

(21) Call subroutine HIST to print and plot time histories of rotational and translational coordinates computed in (14) as well as slopes, etc., computed in (18), if any.

(22) End of loop (12). If all stations of force and displacement analysis have been completed, go to (2) for a new case, otherwise to (13) for the next station.

(23) Close the plot tape and stop.

SUBROUTINE RESCOM

(1) Start loop over all flutter frequencies to form reduced matrices and forcing functions.

(2) Zero the dynamics matrix and call subroutine DF67 with a reduced velocity given by the forward speed and the flutter frequency. Upon return, the complex aerodynamics matrix transpose is provided.

(3) Add the mass matrix transpose to the aerodynamics matrix transpose to form an effective mass matrix.

(4) Call subroutine MULT to form the product of the flexibility transpose and effective mass matrix. The result is the transpose of the system dynamics matrix.

(5) Transpose the dynamics matrix to get in normal column form.

(6) Choose the particular flutter frequency defined by loop (1) and substitute the value for λ in the matrix $A - \lambda I$. In addition, form the complex valued forcing function.

(7) Call subroutine REDUCE to eliminate all unwanted coordinates and retain the 6 x 6 matrix and 6 x 1 force vector necessary to compute determinant values. If all coordinates have not been done, go back to (1).

(8) Call subroutine FN to build the form of the time gust function.

(9) Start loop over all coordinates to compute transfer function coefficients for each flutter frequency and coordinate.

(10) Pick up the reduced matrix corresponding to a particular flutter frequency and substitute the forcing column for the coordinate whose response is desired.

(11) Call subroutine INVDET to evaluate determinant of substituted matrix.

(12) Store coefficients corresponding to a particular coordinate and frequency. If more frequencies remain to be evaluated, return to (10) for another flutter frequency.

(13) Call subroutine RESP to compute the response of a particular coordinate as a linear combination of the responses of each flutter frequency transfer function to the gust input. If all coordinate responses have not been computed, return to (9) for another coordinate.

REDUCE

(1) Set up column subscript vector.

(2) Use loop to divide determinant by denominator factors formed when the first six flutter frequency terms are evaluated at a given flutter frequency.

(3) Start loop over all coordinates to be reduced or eliminated.

(4) Use loop to find the largest diagonal or pivot element remaining of those coordinates which are to be reduced.

(5) If all coordinates to be reduced have been reduced, go to (14).

(6) Set indicator for the coordinate to be reduced to zero, multiply determinant by the pivot element chosen and divide it by the corresponding denominator element evaluated at the given flutter frequency.

(7) Use loop to divide forcing vector and matrix row by the pivot element.

(8) Set the reduced force element to zero to minimize computational error.

(9) Start loop to reduce matrix coordinate.

(10) Skip to (12) if product element is zero or row is the row to be reduced.

(11) Use inner loop to subtract the modified divided row from each row defined by (9).

(12) If all rows have been modified go to (13), otherwise back to (9) for new row.

(13) Use loop to zero the reduced row and column elements. Go to (5) for new coordinate.

(14) Start loop to pick out those elements to be retained for the 6 x 6 matrix.

(15) Pick out all elements of force vector and matrix to be saved and save for column substitution.

(16) Return to subroutine RESCOM.

SUBROUTINE FN

(1) Set the time zero forcing value to zero.

(2) If the gust number is 1 go to (4), if greater than 2 go to (7).

(3) Compute number of points for half cycle sine wave. Gust form is sine wave with input frequency.

(4) Compute sine wave using recursive relationship for sine:

$$\sin(\psi+\Delta\psi) = \sin\psi \cos\Delta\psi + \cos\psi \sin\Delta\psi$$

(5) If the gust number is 1, go to (11).

(6) Use a loop to square the sine wave for gust number 2. Then go to (10).

(7) If the gust number is not equal to 5, go to (9).

(8) Compare Kussner gust function as function of time, forward velocity, and semichord. Then go to (11).

(9) For gust numbers 3 and 4 a step input must be computed. Set the time function to 1. If gust number is 3, go to (11).

(10) For pulse gust and half cycle $\sin^2$ wave the remainder of the response must be set to zero.

(11) Call subroutine TMF to initialize and plot forcing function time response.

(12) Return to subroutine RESCOM.

4. Variable Arrays

(* signifies input parameter)

PC(10,2)	Complex natural frequencies in S domain
FV(74)	Wind force vector amplitudes
Y (1003)	Coordinate response as f(time)
* AWK(74,74)	Flexibility matrix
* G(10)	Flutter damping coefficients
FY(74)	Modified gust forcing vector amplitudes
* Ø(10)	Flutter frequency coefficients
AM(150)	Nonzero mass matrix values
XM(10,6,4)	Storage for transfer function coefficients
YC(1003,6)	Storage for coordinate time responses

C(2,2,22)	Transformation matrices for each linear to displacement rotational coordinate transform
MSB(150)	Nonzero mass matrix indices
NSG(22)	Starting coordinate numbers
NSB(4)	Numbers of coordinate response to be run
NSCB(10)	Coordinate numbers for which response is desired
BUF(512)	Buffer storage for plot output, multiple of 256
TTL(20)	Run title
DY(22)	Section width
B(22)	Section length
XD(1003,3)	Storage for responses to be plotted
EI(10)	EIB section values
GK(10)	GJ section values
SLC(10)	L1 section length values
SLD(10)	L2 section length values
A(74,74)	Complex flutter equations matrix
Z(74)	Work area
* Q(74)	Zero velocity poles or flexible frequencies

APPENDIX I

SAMPLE PROBLEMS

Sample problems are provided to assist the user in checking out and using the computer programs. Each sample problem consists of the data sheets, the input display included in the output, and condensed output. Examples of the CALCOMP machine plots for programs ARAP06 and DFAL18 are also included. The sample problems are taken from the D270A FPR dynamic analysis, discussed in Section IV of Volume I.

IBM **FORTRAN Coding Form** X28-7327-6 U/M050 Printed in U.S.A.

PROGRAM: ARAP06
PROGRAMMER:
DATE:
PUNCHING INSTRUCTIONS: GRAPHIC / PUNCH
PAGE 1 OF 10
CARD ELECTRO NUMBER

Cols 1–10	11–20	21–30	31–40	41–50	51–60	61–70	71–80	
JUNE 11, 1971		SAMPLE CASE	FØR PRØGRAM	ARAP06	USING MØDEL	D270 PARAMETERS		CARD 01
1 1		1 2	1	1 1		1 1		CARD 02
10 2	2 6	6 1						CARD 03
122.	1117.	.002378			160.	160.	386.	CARD 04
3.	-25.	62.9			175.	1.		CARD 05
.0045	0.							CARD 06
				2000.	2000.	2000.	2000.	CARD 07
2000.	2000.	2000.	2000.	50000.	-10.			CARD 08
109.4	.2	1.						CARD 09
600.		10.	1.	1.4	.5	1.44		CARD 10
9.	133.4	5.78	.17	6.	72.2	30.5		CARD 11
0								CARD 12
								CARD 13
								CARD 14
								CARD 15
-2.5	-5.	-7.5	-10.	-12.5	-15.	-17.5	-20.	CARD 16
-22.5	-25.							
30.	60.	90.	120.	150.	180.	210.	240.	CARD 17
270.	300.							
1.5738	.2176	.1344	.1127	.1204	.1325	.1204	.11	CARD 18
.0972	.0435							
0								CARD 19
								CARD 20

ARAP06 Sample Problem, Input Data

IBM FORTRAN Coding Form

PROGRAM ARAP06 PROGRAMMER DATE PUNCHING INSTRUCTIONS GRAPHIC PUNCH PAGE 2 OF 10 CARD ELECTRO NUMBER

X26-7327-6 U/M050 Printed in U.S.A.

```
    0                                                                       CARD 20
                                                                            CARD 21

                                                                            CARD 22

                                                                            CARD 23

40.       40.       40.       40.       40.       40.       40.       40.   CARD 24
40.       40.
15.       30.       30.       30.       30.       30.       30.       30.   CARD 25
30.       15.
40.  52.  64.  76.  90.  105.      7.85 15.7 23.55 28.78 32.86             CARD 26
MØDEL D270  GIMBALLED RØTØR FIRST CØLLECTIVE MØDE                           CARD 27
28.89 31.4 35.8 40.82 49.61 55.89 30.77 33.28 37.05 42.7 50.87 57.15 35.17
37.68 41.45 46.47 54.64 59.66 41.45 43.96 47.73 52.12 59.66 62.8 46.47 48.36 CARD 28
52.12 56.52 62.8 65.31 50.24 52.12 55.26 59.66 65.94 67.2
    1    1    4    0                                                        CARD 29
 .005 .019 .044 .090 .164 .274 .418 .594 .792 1.   .006 .021 .048
 .094 .167 .274 .417 .592 .791 1.   .008 .027 .057 .104 .177 .282
 .421 .594 .792 1.   .010 .036 .072 .123 .195 .297 .432 .600 .794
1.    .016 .055 .106 .167 .245 .344 .471 .627 .807 1.   .025 .087
 .166 .262 .373 .495 .622 .751 .876 1.   .005 .019 .045 .092 .168
 .278 .422 .597 .794 1.   .006 .023 .050 .098 .172 .281 .423 .597
 .793 1.   .008 .028 .059 .108 .182 .287 .427 .598 .794 1.   .011          CARD 30
```

* A standard card form, IBM electro 888157, is available for punching statements from this form

ARAP06 Sample Problem, Input Data (Continued)

IBM FORTRAN Coding Form

X28-7327-6 U/M050 Printed in U.S.A.

PROGRAM ARAP06 | PROGRAMMER | DATE | PUNCHING INSTRUCTIONS: GRAPHIC, PUNCH | PAGE 3 OF 10 | CARD ELECTRO NUMBER

FORTRAN STATEMENT

.037	.075	.127	.201	.303	.438	.605	.797	1.	.017	.058	.111	.175
.254	.355	.481	.635	.812	1.	.026	.088	.168	.264	.374	.494	.635
.746	.874	1.	.006	.022	.051	.103	.182	.294	.438	.610	.801	1.
.007	.026	.057	.108	.186	.297	.439	.609	.8	1.	.009	.032	.066
.119	.196	.304	.443	.611	.801	1.	.012	.042	.083	.140	.218	.322
.457	.619	.804	1.	.019	.064	.121	.191	.274	.377	.503	.652	.821
1.	.026	.090	.171	.267	.375	.493	.617	.744	.872	1.	.007	.027
.06	.117	.201	.316	.46	.626	.81	1.	.009	.03	.066	.123	.206
.318	.459	.625	.809	1.	.011	.037	.077	.136	.218	.328	.466	.628
.81	1.	.014	.049	.096	.158	.241	.348	.481	.638	.815	1.	.021
.072	.137	.213	.303	.41	.535	.677	.835	1.	.027	.091	.173	.270
.377	.494	.617	.743	.871	1.	.008	.03	.067	.128	.216	.333	.476
.639	.816	1.	.01	.035	.074	.135	.222	.337	.477	.639	.816	1.
.012	.042	.085	.148	.234	.346	.483	.642	.817	1.	.016	.054	.105
.172	.258	.366	.498	.651	.822	1.	.023	.077	.145	.226	.319	.427
.551	.69	.842	1.	.027	.092	.175	.271	.379	.495	.617	.743	.871
1.	.009	.033	.073	.138	.229	.348	.489	.649	.822	1.	.011	.038
.08	.145	.234	.35	.489	.648	.821	1.	.013	.045	.092	.158	.246
.359	.495	.651	.822	1.	.017	.058	.112	.183	.271	.381	.512	.662
.827	1.	.024	.081	.151	.235	.33	.44	.563	.7	.847	1.	.027
.093	.175	.272	.38	.497	.62	.746	.873	1.				
-.027	-.052	-.072	-.079	-.072	-.047	-.006	.047	.109	.174	-.039	-.077	-.107
-.12	-.111	-.075	-.014	.07	.168	.272	-.054	-.105	-.147	-.167	-.158	-.11
-.024	.096	.239	.392	-.071	-.14	-.192	-.228	-.219	-.158	-.041	.127	.333

CARD 30

(CONTINUED)

ARAP06 Sample Problem, Input Data (Continued)

IBM FORTRAN Coding Form

PROGRAM: ARAPOL PAGE 4 OF 10

PROGRAMMER: DATE:

FORTRAN STATEMENT

.555	-.096	-.19	-.271	-.321	-.32	-.243	-.078	.171	.488	.633	.108	.212
.295	.335	.318	.225	.053	-.19	-.466	-.804	-.027	-.052	-.072	-.079	-.071
-.046	-.006	.047	.108	.172	-.039	-.076	-.106	-.118	-.109	-.073	-.012	.069
.165	.266	-.053	-.04	-.145	-.165	-.154	-.107	-.022	.095	.235	.384	-.070
-.137	-.193	-.223	-.214	-.153	-.039	.124	.325	.54	-.09	-.178	-.254	-.301
-.299	-.226	-.072	.161	.455	.778	.084	.164	.225	.252	.234	.16	.031
-.146	-.358	-.584	-.026	-.052	-.07	-.077	-.068	-.043	-.004	.046	.104	.164
-.038	-.075	-.103	-.114	-.103	-.063	-.01	.067	.157	.252	-.051	-.101	-.14
-.157	-.145	-.099	-.019	.091	.221	.36	-.066	-.129	-.181	-.207	-.196	-.139
-.033	.117	.299	.495	-.077	-.153	-.217	-.256	-.254	-.191	-.06	.137	.387
.66	.049	.095	.128	.136	.117	.071	.001	-.085	-.18	-.278	-.026	-.051
-.068	-.073	-.063	-.039	-.002	.045	.097	.152	-.037	-.073	-.099	-.108	-.096
-.061	-.006	.065	.147	.233	-.049	-.096	-.131	-.145	-.132	-.088	-.014	.085
.202	.326	-.06	-.117	-.163	-.184	-.172	-.12	-.026	.105	.263	.433	-.059
-.116	-.165	-.193	-.189	-.141	-.043	.104	.288	.49	.032	.061	.079	.079
.061	.029	-.011	-.054	-.096	-.135	-.026	-.050	-.067	-.070	-.06	-.035	
.044	.092	.143	-.036	-.07	-.095	-.102	-.089	-.055	-.004	.062	.137	.216
-.047	-.091	-.124	-.136	-.122	-.079	-.01	.081	.187	.299	-.055	-.109	-.15
-.168	-.155	-.106	-.021	.096	.237	.389	-.049	-.096	-.136	-.159	-.155	-.114
-.034	.086	.236	.401	.027	.051	.066	.063	.047	.019	-.012	-.044	-.074
-.101	-.025	-.049	-.065	-.067	-.056	-.033	.001	.042	.088	.135	-.035	-.069
-.092	-.098	-.084	-.051	-.002	.06	.13	.204	-.045	-.088	-.119	-.128	-.114
-.072	-.008	.077	.175	.28	-.051	-.101	-.139	-.154	-.14	-.094	-.017	.089
.216	.352	-.041	-.082	-.115	-.134	-.13	-.095	-.027	.072	.198	.336	.033

CARD 30

(CONTINUED)

ARAP06 Sample Problem, Input Dat (Continued)

IBM FORTRAN Coding Form

PROGRAM: ARAP06   PAGE 5 OF 10

```
 .063  .082 .084 .069  .039 -.003 -.053 -.109 -.169                       
                                                                  7H    } CARD 30
 60315.   71363.   92130. 125578. 197300. 307509. 63138.  75963.  96842.ROOTBM
 131418. 207380. 312539.   73989.  87932. 110874. 149146. 229406. 321214.
 90239.  105585. 132171. 174519. 261840. 328898. 103772. 121255. 149408.
 193617. 280376. 333980. 115741. 133796. 163145. 209809. 294508. 337462.
                                                                        ROOTCM
                                                                        ROOTCM
                                                                        ROOTCM
                                                                        ROOTCM
 29047.  31062.   34171.   38127.  43043. -53331.  29168.  31255.  34154.BM  AT
 37757.  40793.  -43144.   29707.  31565.  34021.  36587.  35710. -28467.35
 30158.  31663.   33507.   34598.  28331. -20642.  30283.  31578.  32855.
 32879.  24154.  -17913.   30245.  31284.  32136.  31342.  21098. -19417.
-15145. -23292.  -36494.  -56299. -96712.-148271. -15178. -24610. -37899.CM  AT
-58363.-100953. -150925.  -17301. -27540. -42148. -64736.-109557.-154275.35
-19991. -31371.  -48341.  -73357.-121884.-155000. -22212. -35146. -53172.
-79388.-127767. -154409.  -24296. -37707. -56618. -84368.-131796.-152690.   } CARD 31
MODEL D270  GIMBALLED ROTOR SECOND CYCLIC MODE                                CARD 32
81.0   89.18 95.45100.48104.25106.76 39.56 34.54 31.4  28.89 27.60 27.  42.07FR
37.05  33.28 30.77 29.52 28.89 44.59 39.56 35.8  33.28 32.03 31.4  46.47 41.45EQ
38.31  35.17 33.91 33.91 47.73 43.33 39.56 37.05 35.17 35.17              } CARD 33
    1      1     4     1                                                      CARD 34
-.072 -.143 -.219 -.293 -.338 -.309 -.165  .118  .527  1.  -.063 -.125 -.194SE  CARD 35
```

ARAP06 Sample Problem, Input Data (Continued)

IBM

FORTRAN Coding Form

PROGRAM: ARAP06 | PAGE 6 OF 10

-.269	-.324	-.311	-.182	.096	.512	1.	-.056	-.111	-.176	-.25	-.311	-.31	CØ
-.194	.078	.499	1.	-.05	-.097	-.153	-.219	-.276	-.282	-.185	.06	.452	ND
.924	-.027	-.051	-.077	-.105	-.13	-.133	-.09	.026	.219	.457	-.011	-.02	
-.024	-.021	-.017	-.014	-.01	.005	.038	.085	-.067	-.132	-.182	-.205	-.19	CY
-.13	-.025	.118	.289	.472	-.049	-.097	-.135	-.152	-.141	-.097	-.019	.088	CL
.215	.351	-.035	-.068	-.095	-.108	-.101	-.07	-.015	.062	.153	.251	-.022	IC
-.044	-.061	-.07	-.067	-.047	-.011	.039	.101	.168	-.009	-.018	-.026	-.031	
-.031	-.023	-.007	.016	.047	.08	.004	.008	.01	.009	.006	.001	-.003	
-.007	-.009	-.01	-.061	-.12	-.165	-.182	-.165	-.108	-.016	.107	.251	.404	
-.047	-.092	-.126	-.14	-.127	-.084	-.013	.082	.194	.312	-.033	-.066	-.091	
-.101	-.093	-.063	-.011	.059	.142	.23	-.021	-.042	-.059	-.066	-.062	-.043	
-.009	.038	.094	.154	-.009	-.017	-.025	-.029	-.029	-.021	-.006	.016	.043	
.073	.004	.008	.01	.009	.006	.001	-.003	-.007	-.009	-.01	-.054	-.105	
-.141	-.152	-.133	-.082	-.005	.093	.204	.320	-.043	-.084	-.114	-.124	-.109	
-.068	-.006	.075	.167	.263	-.031	-.061	-.084	-.092	-.082	-.053	-.006	.055	
.125	.199	-.02	-.04	-.055	-.061	-.056	-.037	-.006	.036	.085	.137	-.008	
-.016	-.023	-.027	-.026	-.019	-.005	.015	.039	.065	.004	.008	.01	.009	
.006	.001	-.003	-.007	-.009	-.011	-.048	-.094	-.126	-.132	-.112	-.066	.001	
.082	.173	.267	-.04	-.078	-.105	-.112	-.096	-.058	-.001	.069	.147	.229	
-.03	-.053	-.079	-.085	-.074	-.045	-.003	.052	.113	.177	-.019	-.038	-.052	
-.057	-.051	-.032	-.003	.034	.077	.123	-.008	-.015	-.022	-.025	-.023	-.016	
-.004	.014	.035	.059	.004	.008	.01	.009	.006	.001	-.003	-.007	-.01	
-.012	-.044	-.086	-.114	-.118	-.097	-.055	.004	.075	.151	.23	-.038	-.073	
-.098	-.102	-.086	-.05	.002	.064	.133	.205	-.028	-.055	-.075	-.079	-.067	

CARD 35

(CONTINUED)

ARAP06 Sample Problem, Input Data (Continued)

IBM FORTRAN Coding Form

PROGRAM ARAP06 PAGE 7 OF 10

PROGRAMMER DATE PUNCHING INSTRUCTIONS GRAPHIC PUNCH CARD ELECTRO NUMBER

FORTRAN STATEMENT

-.04	-.001	.049	.104	.162	-.018	-.036	-.05	-.053	-.047	-.029	-.002	.032
.071	.113	-.007	-.014	-.02	-.023	-.022	-.015	-.003	.013	.033	.053	.004
.009	.01	.009	.006	.001	-.003	-.007	-.01	-.012	.			52
-.021	-.072	-.139	-.215	-.284	-.323	-.313	-.249	-.14	-.011	-.023	-.077	-.155X2
-.251	-.339	-.38	-.341	-.204	.02	.283	-.026	-.089	-.185	-.312	-.434	-.49
-.423	-.196	.175	.616	-.03	-.103	-.223	-.389	-.554	-.633	-.537	-.206	.342
1.	-.024	-.085	-.186	-.333	-.484	-.559	-.472	-.16	.366	1.	-.023	-.08
-.176	-.316	-.462	-.537	-.454	-.15	.369	1.	.011	.039	.083	.149	.24
.358	.499	.657	.827	1.	.008	.029	.065	.125	.212	.329	.473	.638
.816	1.	.007	.024	.055	.111	.195	.312	.457	.626	.81	1.	.006
.02	.049	.102	.184	.299	.446	.617	.805	1.	.005	.018	.045	.096
.176	.291	.438	.611	.802	1.	.005	.018	.043	.092	.171	.284	.43
.605	.798	1.	.013	.045	.094	.165	.26	.38	.519	.673	.835	1.
.01	.034	.074	.139	.231	.35	.493	.653	.825	1.	.008	.027	.063
.123	.211	.33	.475	.64	.817	1.	.007	.023	.055	.113	.199	.317
.463	.631	.813	1.	.006	.021	.051	.106	.19	.307	.453	.623	.808
1.	.006	.02	.049	.102	.184	.3	.446	.617	.805	1.	.015	.053
.108	.186	.286	.408	.545	.692	.845	1.	.012	.04	.087	.158	.255
.377	.518	.672	.835	1.	.009	.033	.073	.14	.234	.356	.499	.658
.827	1.	.008	.028	.065	.128	.22	.34	.485	.647	.821	1.	.007
.025	.059	.12	.209	.329	.474	.639	.817	1.	.007	.024	.058	.116
.204	.322	.467	.634	.814	1.	.017	.058	.117	.2	.303	.425	.56
.704	.851	1.	.013	.045	.096	.172	.273	.396	.535	.685	.842	1.
.011	.037	.082	.153	.257	.374	.516	.671	.834	1.	.009	.032	.073

*A standard card form, IBM electro 888157, is available for punching statements from this form

CARD 35

(CONTINUED)

ARAP06 Sample Problem, Input Data (Continued)

IBM FORTRAN Coding Form

PROGRAM ARAP06 | PROGRAMMER | DATE | PUNCHING INSTRUCTIONS | GRAPHIC | PUNCH | PAGE 8 OF 10 | CARD ELECTRO NUMBER

CARD 35

.141	.236	.359	.502	.661	.829	1.	.008	.028	.066	.131	.225	.346
.49	.652	.824	1.	.008	.028	.064	.127	.219	.339	.483	.645	.82
1.	.018	.062	.124	.21	.315	.438	.571	.711	.855	1.	.014	.049
.103	.182	.285	.409	.547	.694	.846	1.	.012	.04	.088	.163	.264
.388	.528	.68	.839	1.	.01	.035	.079	.15	.249	.372	.515	.67
.834	1.	.009	.031	.072	.141	.237	.36	.503	.661	.829	1.	.009
.031	.07	.137	.231	.352	.495	.655	.825	1.				X2
												TH

CARD 36

									RØØTBM
									RØØTBM
									RØØTBM
									RØØTBM
-243744.	-260724.	-296885.	-344684.	-282371.	-269814.	127877.	93318.	73468.	RØØTCM
61777.	55278.	53663.	147888.	109018.	85807.	72534.	64667.	62827.	
174762.	131499.	104585.	88463.	79048.	77305.	193754.	148644.	119596.	
101969.	91220.	89098.	208213.	162130.	131867.	113112.	101763.	99532.	
8883.	-3221.	-16666.	-31267.	-32854.	-33336.	41891.	36499.	32607.	BM AT
29914.	28178.	27495.	40283.	36258.	32840.	30444.	28744.	28065.	35
37661.	35437.	32853.	30766.	29243.	28675.	35491.	34488.	32573.	BM AT
30858.	29463.	28873.	33641.	33477.	32103.	30683.	29486.	28928.	35
-118814.	-117055.	-114283.	-103125.	-49901.	-7353.	51459.	31883.	19575.	CM AT
11144.	3847.	-2882.	59854.	37630.	23328.	13447.	4772.	-3241.	35
70437.	45469.	28678.	16279.	6084.	-3822.	77271.	51013.	32647.	CM AT
19258.	7092.	-4330.	82068.	55183.	35736.	21248.	7894.	-4768.	35

*A standard card form, IBM electro 888157, is available for punching statements from this form

CARDS 37 → 46
ARE DELETED
FOR THIS RUN.

ARAP06 Sample Problem, Input Data (Continued)

IBM FORTRAN Coding Form

Program: ARAP06 — Page 9 of 10

FORTRAN STATEMENT

Cols 1–10	11–20	21–30	31–40	41–50	51–60	61–70	71–80	
								CARD 47
8 4	3							CARD 48
.01	7.9	83.8			.01	9.35	16390.	CARD 49
-1.1	.258	-.73	-.37					CARD 50
.0102	.0107	.0148	.0172	.0203	.0252	.0265	.0338	CARD 51
.3187	.294	.1703	.0789	-.0517	-.2973	-.3669	-.7963	
-.0204	-.0208	-.0271	-.0318	-.0396	-.0574	-.0632	-.1051	
-.009	-.0062	.0018	.0046	.0044	-.0074	-.0134	-.0704	
.3693	.3427	.211	.1109	-.0335	-.3171	-.3975	-.881	CARD 52
.0255	.0227	.0108	.0014	-.0119	-.036	-.0426	-.0771	
-.097	-.066	.0635	.1485	.2544	.4072	.4404	.5626	
-.2924	-.3116	-.3507	-.3389	-.2772	-.0679	.0051	.4916	
		.0001	.0003	.0007	.001	.0014	.0017	CARD 53
	-.0001	-.0003	-.0005	-.0007	-.0009	-.0011	-.0012	
	.0014	.0027	.004	.0053	.0067	.008	.0093	
.0003	.0014	.0025	.0036	.0047	.0058	.0069	.008	
28.0434	8.11	6.55	8.18	4.89	7.38	3.57	8010	CARD 54
			15760.		15760.			
	37.5	55.	37.5	50.	37.5	47.5	40.	
17.48	33.06	40.69	60.64	.0361	-1.	-.1316	-.11	CARD 55
				-1.	-.1323	1.	1.	
-.00026	.0065	.0001	.00077	.0038	-.0014	.0151	.0098	
13.2503	16.2371	9.8209	3.0095					

ARAP06 Sample Problem, Input Data (Continued)

IBM FORTRAN Coding Form

PROGRAM: ARAP06 | PROGRAMMER: | DATE: | PAGE 10 OF 10

STATEMENT NUMBER | CONT. | FORTRAN STATEMENT | IDENTIFICATION SEQUENCE

CARD 56

CARD 57

```
                                                  83.8
103.
```

ARAP06 Sample Problem, Input Data (Concluded)

```
                     ARAP06  FOLDING PROPROTOR FEATHER-STOP DYNAMIC ANALYSIS

TITLE            JUNE 11, 1971   SAMPLE CASE FOR PROGRAM ARAP06 USING MODEL D270 PARAMETERS

FLIGHT CONDITION
U = 175.00 KTS           RPM = 160.0              G = 386.0 IN/SEC**2      GV =   0.0 IN/SEC**2      S = 1117. FT/SEC
TPC =  0.0  DEG          RHO = 0.002378 SLUG/FT**3

OPTIONS AND CONTROLS
IT1 = 1                  IT2 = 1                  IT3 = 0                  IT4 = 0                  IT5 = 1
IT6 = 2                  IT7 = 1                  IT8 = 0                  IT9 = 1                  IT10 = 1
IT11 = 0                 IT12 = 0                 IT13 = 1                 IT14 = 1                 IT15 = 0
NCASE = 1                DTMIN = 0.0045 SEC       PSIE =   0.0 DEG
RPM1 =   0.0 RPM         RPM2 =   0.0 RPM         RPM3 =   0.0 RPM         RPM4 =   0.0 RPM         AK11 =  2000.0 FT-LB/DEG
AK12 =  2000.0 FT-LB/DEG AK13 =  2000.0 FT-LB/DEG AK14 =  2000.0 FT-LB/DEG BK11 =  2000.0 FT-LB/DEG BK12 =  2000.0 FT-LB/DEG
BK13 =  2000.0 FT-LB/DEG BK14 =  2000.0 FT-LB/DEG STOPK = 50000.0 FT-LB/DEG STOPTH = -10.0 DEG
ALW =   9.00 DEG         WC = 133.40 IN           SLC = 5.780
DEL = 0.17               ALAM =   6.00 DEG        AL = 72.20 IN            YCA =  30.50 IN

PROPROTOR DATA
NS = 10                  A0 =   3.0 DEG           DEL3 = -25.00 DEG        THC = 62.90 DEG          A1 =  0.0  DEG
B1 =  0.0  DEG           NBM = 2                  BZETA = 0.0              OMT = 160.00 RAD/SEC     ZM = 122.0 IN
NMT = 2                  NTHC = 6                 NPSID = 6
```

SEGMENT	THT(I) DEG	R(I) IN	AM(I) CHUG	YB(I) IN	AIC(I) CHUG-IN.SQ	EPF(I) IN	EPC(I) IN	EPE(I) IN	Z(I) IN	DR(I) IN
1	-2.50	30.0	1.5738	0.0	0.0	0.0	0.0	0.0	40.00	15.0
2	-5.00	60.0	0.2176	0.0	0.0	0.0	0.0	0.0	40.00	30.0
3	-7.50	90.0	0.1344	0.0	0.0	0.0	0.0	0.0	40.00	30.0
4	-10.00	120.0	0.1127	0.0	0.0	0.0	0.0	0.0	40.00	30.0
5	-12.50	150.0	0.1204	0.0	0.0	0.0	0.0	0.0	40.00	30.0
6	-15.00	180.0	0.1325	0.0	0.0	0.0	0.0	0.0	40.00	30.0
7	-17.50	210.0	0.1204	0.0	0.0	0.0	0.0	0.0	40.00	30.0
8	-20.00	240.0	0.1100	0.0	0.0	0.0	0.0	0.0	40.00	30.0
9	-22.50	270.0	0.0972	0.0	0.0	0.0	0.0	0.0	40.00	30.0
10	-25.00	300.0	0.0435	0.0	0.0	0.0	0.0	0.0	40.00	15.0

ARAP06 Sample Problem, Input Display

BELL HELICOPTER COMPANY

PROPROTOR BLADE NORMAL MODES

MODEL D270 GIMBALLED ROTOR FIRST COLLECTIVE MODE

KDR(1) = 0

NATURAL FREQUENCY

THC (DEG)	ROTOR RPM (RAD/SEC) 0.0	7.8	15.7	23.5	28.8	32.9
40.0	28.9	30.8	35.2	41.4	46.5	50.2
52.0	31.4	33.3	37.7	44.0	48.4	52.1
64.0	35.8	37.0	41.4	47.7	52.1	55.3
76.0	40.8	42.7	46.5	52.1	56.5	59.7
90.0	49.6	50.9	54.6	59.7	62.8	65.9
105.0	55.9	57.1	59.7	62.8	65.3	67.2

OUT-OF-PLANE MODE SHAPE

LC(1) = 1

RPM = 0.0 RAD/SEC

THC \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	0.005	0.019	0.044	0.090	0.164	0.274	0.418	0.594	0.792	1.000
52.0	0.006	0.021	0.048	0.094	0.167	0.274	0.417	0.592	0.791	1.000
64.0	0.008	0.027	0.057	0.104	0.177	0.282	0.421	0.594	0.792	1.000
76.0	0.010	0.036	0.072	0.123	0.195	0.297	0.432	0.600	0.794	1.000
90.0	0.016	0.055	0.106	0.167	0.249	0.344	0.471	0.627	0.807	1.000
105.0	0.025	0.087	0.166	0.262	0.373	0.495	0.622	0.751	0.876	1.000

RPM = 7.8 RAD/SEC

THC \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	0.005	0.019	0.045	0.092	0.168	0.278	0.422	0.597	0.794	1.000
52.0	0.006	0.023	0.050	0.098	0.172	0.281	0.423	0.597	0.793	1.000
64.0	0.008	0.028	0.059	0.108	0.182	0.287	0.427	0.598	0.794	1.000
76.0	0.011	0.037	0.075	0.127	0.201	0.303	0.438	0.605	0.797	1.000
90.0	0.017	0.058	0.111	0.175	0.254	0.355	0.481	0.635	0.812	1.000
105.0	0.026	0.088	0.168	0.264	0.374	0.494	0.620	0.748	0.874	1.000

RPM = 15.7 RAD/SEC

THC \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	0.006	0.022	0.051	0.103	0.182	0.294	0.438	0.610	0.801	1.000
52.0	0.007	0.026	0.057	0.108	0.186	0.297	0.439	0.609	0.800	1.000
64.0	0.009	0.032	0.066	0.119	0.196	0.304	0.443	0.611	0.801	1.000
76.0	0.012	0.042	0.083	0.140	0.218	0.322	0.457	0.619	0.804	1.000
90.0	0.019	0.064	0.121	0.191	0.274	0.377	0.503	0.652	0.821	1.000
105.0	0.026	0.090	0.171	0.267	0.375	0.493	0.617	0.744	0.872	1.000

RPM = 23.5 RAD/SEC

THC \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	0.007	0.027	0.060	0.117	0.201	0.316	0.460	0.626	0.810	1.000
52.0	0.009	0.030	0.066	0.123	0.206	0.318	0.459	0.625	0.809	1.000
64.0	0.011	0.037	0.077	0.136	0.218	0.328	0.466	0.628	0.810	1.000
76.0	0.014	0.049	0.096	0.158	0.241	0.348	0.481	0.638	0.815	1.000
90.0	0.021	0.072	0.137	0.213	0.305	0.418	0.535	0.677	0.835	1.000
105.0	0.027	0.091	0.173	0.270	0.377	0.494	0.617	0.743	0.871	1.000

ARAP06 Sample Problem, Input Display (Continued)

RPM = 28.8 RAD/SEC

T/C \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	0.008	0.030	0.067	0.128	0.216	0.333	0.476	0.639	0.816	1.000
52.0	0.010	0.035	0.074	0.135	0.222	0.337	0.477	0.639	0.816	1.000
64.0	0.012	0.042	0.085	0.148	0.234	0.346	0.483	0.642	0.817	1.000
76.0	0.016	0.054	0.105	0.172	0.258	0.366	0.498	0.651	0.822	1.000
90.0	0.023	0.077	0.145	0.226	0.319	0.427	0.551	0.690	0.842	1.000
105.0	0.027	0.092	0.175	0.271	0.379	0.495	0.617	0.743	0.871	1.000

RPM = 32.9 RAD/SEC

T/C \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	0.009	0.033	0.073	0.138	0.229	0.348	0.489	0.649	0.822	1.000
52.0	0.011	0.038	0.080	0.145	0.234	0.350	0.489	0.648	0.821	1.000
64.0	0.013	0.045	0.092	0.158	0.246	0.359	0.495	0.651	0.822	1.000
76.0	0.017	0.058	0.112	0.183	0.271	0.381	0.512	0.662	0.827	1.000
90.0	0.024	0.081	0.151	0.235	0.330	0.440	0.563	0.700	0.847	1.000
105.0	0.027	0.093	0.175	0.272	0.380	0.497	0.620	0.746	0.873	1.005

INPLANE MODE SHAPE

LC(2) = 1

RPM = 0.0 RAD/SEC

T/C \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.027	-0.052	-0.072	-0.079	-0.072	-0.047	-0.006	0.047	0.109	0.174
52.0	-0.039	-0.077	-0.107	-0.120	-0.111	-0.075	-0.014	0.070	0.168	0.272
64.0	-0.054	-0.105	-0.147	-0.167	-0.158	-0.110	-0.024	0.096	0.239	0.392
76.0	-0.071	-0.140	-0.192	-0.228	-0.219	-0.158	-0.041	0.127	0.333	0.555
90.0	-0.096	-0.190	-0.271	-0.321	-0.320	-0.243	-0.078	0.171	0.488	0.833
105.0	0.108	0.212	0.295	0.335	0.318	0.225	0.053	-0.190	-0.486	-0.804

RPM = 7.8 RAD/SEC

T/C \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.027	-0.052	-0.072	-0.079	-0.071	-0.046	-0.006	0.047	0.108	0.172
52.0	-0.039	-0.076	-0.106	-0.118	-0.109	-0.073	-0.012	0.069	0.165	0.266
64.0	-0.053	-0.104	-0.145	-0.165	-0.154	-0.107	-0.022	0.095	0.235	0.384
76.0	-0.070	-0.137	-0.193	-0.223	-0.214	-0.153	-0.039	0.124	0.325	0.540
90.0	-0.090	-0.178	-0.254	-0.301	-0.299	-0.226	-0.077	0.161	0.455	0.778
105.0	0.084	0.164	0.225	0.252	0.234	0.160	0.031	-0.146	-0.358	-0.584

RPM = 15.7 RAD/SEC

T/C \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.026	-0.052	-0.070	-0.077	-0.068	-0.043	-0.004	0.046	0.104	0.164
52.0	-0.038	-0.075	-0.103	-0.114	-0.103	-0.068	-0.010	0.067	0.157	0.252
64.0	-0.051	-0.101	-0.140	-0.157	-0.145	-0.099	-0.019	0.091	0.221	0.360
76.0	-0.066	-0.129	-0.181	-0.207	-0.196	-0.139	-0.033	0.117	0.299	0.495
90.0	-0.077	-0.153	-0.217	-0.256	-0.254	-0.191	-0.060	0.137	0.387	0.660
105.0	0.049	0.095	0.128	0.136	0.117	0.071	0.001	-0.085	-0.180	-0.278

RPM = 23.5 RAD/SEC

T/C \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.026	-0.051	-0.068	-0.073	-0.063	-0.039	-0.002	0.045	0.097	0.152
52.0	-0.037	-0.073	-0.099	-0.108	-0.096	-0.061	-0.006	0.065	0.147	0.233
64.0	-0.049	-0.096	-0.131	-0.145	-0.132	-0.088	-0.014	0.085	0.202	0.326
76.0	-0.060	-0.117	-0.163	-0.184	-0.172	-0.120	-0.026	0.105	0.263	0.433
90.0	-0.059	-0.116	-0.165	-0.193	-0.189	-0.141	-0.043	0.104	0.288	0.490
105.0	[illegible]	0.061	0.079	0.079	0.061	0.029	-0.011	-0.054	-0.096	-0.135

ARAP06 Sample Problem, Input Display (Continued)

RPM = 28.8 RAD/SEC

THC \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.026	-0.050	-0.067	-0.070	-0.060	-0.035	0.0	0.044	0.092	0.143
52.0	-0.036	-0.070	-0.095	-0.102	-0.089	-0.055	-0.006	0.062	0.137	0.216
64.0	-0.047	-0.091	-0.124	-0.136	-0.122	-0.079	-0.010	0.081	0.187	0.299
76.0	-0.055	-0.109	-0.150	-0.168	-0.155	-0.106	-0.021	0.096	0.237	0.389
90.0	-0.049	-0.096	-0.136	-0.159	-0.155	-0.114	-0.034	0.086	0.236	0.4[illegible]
105.0	0.027	0.051	0.066	0.063	0.047	0.019	-0.012	-0.044	-0.074	-0.101

RPM = 32.9 RAD/SEC

THC \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.025	-0.049	-0.065	-0.067	-0.056	-0.033	0.001	0.042	0.088	0.135
52.0	-0.035	-0.069	-0.092	-0.098	-0.084	-0.051	-0.002	0.060	0.130	0.204
64.0	-0.045	-0.088	-0.119	-0.128	-0.114	-0.072	-0.008	0.077	0.175	0.280
76.0	-0.051	-0.101	-0.139	-0.154	-0.140	-0.094	-0.017	0.089	0.216	0.352
90.0	-0.041	-0.082	-0.115	-0.134	-0.130	-0.095	-0.027	0.072	0.198	0.336
105.0	0.033	0.063	0.082	0.084	0.069	0.039	-0.003	-0.053	-0.109	-0.169

TORSIONAL MODE SHAPE

LC(3) = 4 FOR ALL THC AND RPM

STATION	1	2	3	4	5	6	7	8	9	10
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

BEAM MOMENT AT SPECIFIED STATION 1

THC (DEG)	ROTOR RPM (RAD/SEC) 0.0	7.8	15.7	23.5	28.8	32.9
40.0	60315.0	63138.0	73989.0	90239.0	103772.0	115741.0
52.0	71363.0	75963.0	87932.0	105585.0	121255.0	133796.0
64.0	92130.C	96842.C	110874.0	132171.0	149408.0	163145.0
76.0	125578.0	131418.C	149146.0	174519.0	193617.0	209809.0
90.0	197300.0	207380.C	229406.0	261840.0	280376.0	294508.0
105.0	307509.0	312539.0	321214.0	328898.0	333980.0	337462.0

CHORD MOMENT AT SPECIFIED STATION 1

THC (DEG)	ROTOR RPM (RAD/SEC) 0.0	7.8	15.7	23.5	28.8	32.9
40.0	0.0	0.0	0.0	0.0	0.0	0.0
52.0	0.0	0.C	0.0	0.0	0.0	0.0
64.0	0.0	C.0	0.0	0.0	0.0	0.0
76.0	0.0	0.0	0.0	0.0	0.0	0.0
90.0	0.0	0.0	0.0	0.0	0.0	0.0
105.0	0.0	0.0	0.0	0.0	0.0	0.0

ARAP06 Sample Problem, Input Display (Continued)

BEAM MOMENT AT SPECIFIED STATION 2

THC (DEG)	ROTOR RPM (RAD/SEC) 0.0	7.8	15.7	23.5	28.8	32.9
40.0	29047.0	29168.0	29767.0	30158.0	30283.0	30245.0
52.0	31062.0	31255.0	31565.0	31663.0	31578.0	31284.0
64.0	34171.0	34154.0	34021.0	33507.0	32855.0	32136.0
76.0	38127.0	37757.0	36587.0	34598.0	32979.0	31342.0
90.0	43043.0	40393.0	35710.0	28331.0	24154.0	21098.0
105.0	-53331.0	-43144.0	-28467.0	-20642.0	-17913.0	-19417.0

CHORD MOMENT AT SPECIFIED STATION 2

THC (DEG)	ROTOR RPM (RAD/SEC) 0.0	7.8	15.7	23.5	28.8	32.9
40.0	-15145.0	-15178.0	-17301.0	-19991.0	-22212.0	-24296.0
52.0	-23292.0	-24610.0	-27540.0	-31371.0	-35146.0	-37707.0
64.0	-36494.0	-37899.0	-42148.0	-48341.0	-53172.0	-56618.0
76.0	-56299.0	-58363.0	-64736.0	-73357.0	-79388.0	-84368.0
90.0	-96712.0	-100953.0	-109557.0	-121884.0	-127767.0	-131796.0
105.0	-148271.0	-150925.0	-154275.0	-155000.0	-154409.0	-152690.0

MODEL 0270 GIMBALLED ROTOR SECOND CYCLIC MODE

KDR(2) = 1

NATURAL FREQUENCY

THC (DEG)	ROTOR RPM (RAD/SEC) 0.0	7.8	15.7	23.5	28.8	32.9
40.0	81.0	39.6	42.1	44.6	46.5	47.7
52.0	89.2	34.5	37.0	39.6	41.4	43.3
64.0	95.4	31.4	33.3	35.8	38.3	39.6
76.0	100.5	28.9	30.0	33.3	35.2	37.0
90.0	104.2	27.6	29.5	32.0	33.9	35.2
105.0	106.8	27.0	28.9	31.4	33.9	35.2

OUT-OF-PLANE MODE SHAPE

LC(1) = 1
RPM = 0.0 RAD/SEC

STATION / THC	1	2	3	4	5	6	7	8	9	10
40.0	-0.072	-0.143	-0.219	-0.293	-0.338	-0.309	-0.165	0.118	0.527	1.000
52.0	-0.063	-0.125	-0.194	-0.269	-0.324	-0.311	-0.182	0.096	0.512	1.000
64.0	-0.056	-0.111	-0.176	-0.250	-0.311	-0.310	-0.194	0.078	0.499	1.000
76.0	-0.050	-0.097	-0.153	-0.219	-0.276	-0.282	-0.185	0.060	0.452	0.924
90.0	-0.027	-0.051	-0.077	-0.105	-0.130	-0.133	-0.090	0.026	0.219	0.457
105.0	-0.011	-0.020	-0.024	-0.021	-0.017	-0.014	-0.010	0.005	0.038	0.085

BELL HELICOPTER COMPANY

ARAP06 Sample Problem, Input Display (Continued)

RPM = 7.8 RAD/SEC

T-C \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.067	-0.132	-0.182	-0.205	-0.190	-0.130	-0.025	0.118	0.289	0.472
52.0	-0.049	-0.097	-0.135	-0.152	-0.141	-0.097	-0.019	0.088	0.215	0.351
64.0	-0.035	-0.068	-0.095	-0.108	-0.101	-0.070	-0.015	0.062	0.153	0.251
76.0	-0.022	-0.044	-0.061	-0.070	-0.067	-0.047	-0.011	0.039	0.101	0.168
90.0	-0.009	-0.018	-0.026	-0.031	-0.031	-0.023	-0.007	0.016	0.047	0.080
105.0	0.004	0.008	0.010	0.009	0.006	0.001	-0.003	-0.007	-0.009	-0.010

RPM = 15.7 RAD/SEC

T-C \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.061	-0.120	-0.165	-0.182	-0.165	-0.108	-0.016	0.107	0.251	0.404
52.0	-0.047	-0.092	-0.126	-0.140	-0.127	-0.084	-0.013	0.082	0.194	0.312
64.0	-0.033	-0.066	-0.091	-0.101	-0.093	-0.063	-0.011	0.059	0.142	0.230
76.0	-0.021	-0.042	-0.059	-0.066	-0.062	-0.043	-0.009	0.038	0.094	0.154
90.0	-0.009	-0.017	-0.025	-0.029	-0.029	-0.021	-0.006	0.016	0.043	0.073
105.0	0.004	0.008	0.010	0.009	0.006	0.001	-0.003	-0.007	-0.009	-0.010

RPM = 23.5 RAD/SEC

T-C \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.054	-0.105	-0.141	-0.152	-0.133	-0.082	-0.005	0.093	0.204	0.320
52.0	-0.043	-0.084	-0.114	-0.124	-0.109	-0.068	-0.006	0.075	0.167	0.263
64.0	-0.031	-0.061	-0.084	-0.092	-0.082	-0.053	-0.006	0.055	0.125	0.199
76.0	-0.020	-0.040	-0.055	-0.061	-0.056	-0.037	-0.006	0.036	0.085	0.137
90.0	-0.008	-0.016	-0.023	-0.027	-0.026	-0.019	-0.005	0.015	0.039	0.065
105.0	0.004	0.008	0.010	0.009	0.006	0.001	-0.003	-0.007	-0.009	-0.011

BELL HELICOPTER COMPANY

RPM = 28.0 RAD/SEC

T-C \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.048	-0.094	-0.126	-0.132	-0.112	-0.066	0.001	0.082	0.173	0.267
52.0	-0.040	-0.078	-0.105	-0.112	-0.096	-0.058	-0.001	0.069	0.147	0.229
64.0	-0.030	-0.058	-0.079	-0.085	-0.074	-0.045	-0.003	0.052	0.113	0.177
76.0	-0.019	-0.038	-0.052	-0.057	-0.051	-0.032	-0.003	0.034	0.077	0.123
90.0	-0.008	-0.015	-0.022	-0.025	-0.023	-0.016	-0.004	0.014	0.035	0.059
105.0	0.004	0.008	0.010	0.009	0.006	0.001	-0.003	-0.007	-0.010	-0.012

RPM = 32.9 RAD/SEC

T-C \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.044	-0.086	-0.114	-0.118	-0.097	-0.055	0.004	0.075	0.151	0.230
52.0	-0.038	-0.073	-0.098	-0.102	-0.086	-0.050	0.002	0.064	0.133	0.205
64.0	-0.028	-0.055	-0.075	-0.079	-0.067	-0.040	-0.001	0.049	0.104	0.162
76.0	-0.018	-0.036	-0.050	-0.053	-0.047	-0.029	-0.002	0.032	0.071	0.113
90.0	-0.007	-0.014	-0.020	-0.023	-0.022	-0.015	-0.003	0.013	0.033	0.053
105.0	0.004	0.008	0.010	0.009	0.006	0.001	-0.003	-0.007	-0.010	-0.012

ARAP06 Sample Problem, Input Display (Continued)

INPLANE MODE SHAPE

LC(2) = 1

RPM = 0.0 RAD/SEC

THC \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	-0.021	-0.072	-0.139	-0.215	-0.284	-0.373	-0.313	-0.249	-0.140	-0.011
52.0	-0.023	-0.077	-0.155	-0.251	-0.339	-0.380	-0.341	-0.204	0.020	0.283
64.0	-0.026	-0.089	-0.185	-0.312	-0.434	-0.490	-0.423	-0.196	0.175	0.616
76.0	-0.030	-0.103	-0.223	-0.389	-0.554	-0.633	-0.537	-0.206	0.342	1.000
90.0	-0.024	-0.085	-0.186	-0.333	-0.484	-0.559	-0.472	-0.160	0.366	1.000
105.0	-0.023	-0.080	-0.176	-0.316	-0.462	-0.537	-0.454	-0.150	0.369	1.000

RPM = 7.8 RAD/SEC

THC \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	0.011	0.039	0.083	0.149	0.240	0.358	0.499	0.657	0.827	1.000
52.0	0.008	0.029	0.065	0.125	0.212	0.329	0.473	0.638	0.816	1.000
64.0	0.007	0.024	0.055	0.111	0.195	0.312	0.457	0.626	0.810	1.000
76.0	0.006	0.020	0.049	0.102	0.184	0.299	0.446	0.617	0.805	1.000
90.0	0.005	0.018	0.045	0.096	0.176	0.291	0.438	0.611	0.802	1.000
105.0	0.005	0.018	0.043	0.092	0.171	0.284	0.430	0.605	0.798	1.000

RPM = 15.7 RAD/SEC

THC \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	0.013	0.045	0.094	0.165	0.260	0.380	0.519	0.673	0.835	1.000
52.0	0.010	0.034	0.074	0.139	0.231	0.350	0.493	0.653	0.825	1.000
64.0	0.008	0.027	0.063	0.123	0.211	0.330	0.475	0.640	0.817	1.000
76.0	0.007	0.023	0.055	0.113	0.199	0.317	0.463	0.631	0.813	1.000
90.0	0.006	0.021	0.051	0.106	0.190	0.307	0.453	0.623	0.808	1.000
105.0	0.006	0.020	0.049	0.102	0.184	0.300	0.446	0.617	0.805	1.000

RPM = 23.5 RAD/SEC

THC \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	0.015	0.053	0.108	0.186	0.286	0.408	0.545	0.692	0.845	1.000
52.0	0.012	0.040	0.087	0.158	0.255	0.377	0.518	0.672	0.835	1.000
64.0	0.009	0.033	0.073	0.140	0.234	0.356	0.499	0.658	0.827	1.000
76.0	0.008	0.028	0.065	0.128	0.220	0.340	0.485	0.647	0.821	1.000
90.0	0.007	0.025	0.059	0.120	0.209	0.329	0.474	0.639	0.817	1.000
105.0	0.007	0.024	0.058	0.116	0.204	0.322	0.467	0.634	0.814	1.000

RPM = 28.8 RAD/SEC

THC \ STATION	1	2	3	4	5	6	7	8	9	10
40.0	0.017	0.058	0.117	0.200	0.303	0.425	0.560	0.704	0.851	1.000
52.0	0.013	0.045	0.096	0.172	0.273	0.396	0.535	0.685	0.842	1.000
64.0	0.011	0.037	0.082	0.153	0.257	0.374	0.516	0.671	0.834	1.000
76.0	0.009	0.032	0.073	0.141	0.236	0.359	0.502	0.661	0.829	1.000
90.0	0.008	0.028	0.066	0.131	0.225	0.346	0.490	0.652	0.824	1.000
105.0	0.008	0.028	0.064	0.127	0.219	0.339	0.483	0.645	0.822	1.000

BELL HELICOPTER COMPANY

ARAP06 Sample Problem, Input Display (Continued)

RPM = 32.9 RAD/SEC

THC	STATION 1	2	3	4	5	6	7	8	9	10
40.0	0.018	0.062	0.124	0.210	0.315	0.438	0.571	0.711	0.855	1.000
52.0	0.014	0.049	0.103	0.182	0.285	0.409	0.547	0.694	0.846	1.000
64.0	0.012	0.040	0.088	0.163	0.264	0.388	0.528	0.680	0.839	1.000
76.0	0.010	0.035	0.079	0.150	0.249	0.372	0.515	0.670	0.834	1.000
90.0	0.009	0.031	0.072	0.141	0.237	0.360	0.503	0.661	0.829	1.000
105.0	0.009	0.031	0.070	0.137	0.231	0.352	0.495	0.655	0.825	1.000

TORSIONAL MODE SHAPE

LC(3) = 4 FOR ALL THC AND RPM

STATION 1	2	3	4	5	6	7	8	9	10
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

BEAM MOMENT AT SPECIFIED STATION 1

THC (DEG)	ROTOR RPM (RAD/SEC) 0.0	7.8	15.7	23.5	28.8	32.9
40.0	0.0	0.0	0.0	0.0	0.0	0.0
52.0	0.0	0.0	0.0	0.0	0.0	0.0
64.0	0.0	0.0	0.0	0.0	0.0	0.0
76.0	0.0	0.0	0.0	0.0	0.0	0.0
90.0	0.0	0.0	0.0	0.0	0.0	0.0
105.0	0.0	0.0	0.0	0.0	0.0	0.0

CHORD MOMENT AT SPECIFIED STATION 1

THC (DEG)	ROTOR RPM (RAD/SEC) 0.0	7.8	15.7	23.5	28.8	32.9
40.0	-243744.0	127877.0	147888.0	174762.0	193754.0	208213.0
52.0	-260724.0	93318.0	109018.0	131499.0	148644.0	167130.0
64.0	-296885.0	73468.0	85807.0	104585.0	119596.0	131867.0
76.0	-344684.0	61777.0	72534.0	88463.0	101969.0	113112.0
90.0	-282371.0	55278.0	64667.0	79048.0	91220.0	101763.0
105.0	-269814.0	53663.0	62827.0	77305.0	89098.0	99532.0

BEAM MOMENT AT SPECIFIED STATION 2

THC (DEG)	ROTOR RPM (RAD/SEC) 0.0	7.8	15.7	23.5	28.8	32.9
40.0	8883.0	41891.0	40283.0	37661.0	35491.0	33641.0
52.0	-3221.0	36499.0	36258.0	35437.0	34488.0	33477.0
64.0	-16666.0	32607.0	32840.0	32853.0	32573.0	32103.0
76.0	-31267.0	29914.0	30444.0	30766.0	30858.0	30693.0
90.0	-32854.0	28170.0	28744.0	29243.0	29463.0	29496.0
105.0	-33336.0	27495.0	28065.0	28675.0	28873.0	28978.0

ARAP06 Sample Problem, Input Display (Continued)

CHORD MOMENT AT SPECIFIED STATION 2

THC (DEG)	ROTOR RPM (RAD/SEC) 0.0	7.0	15.7	23.5	28.0	32.9
40.0	-118814.0	51459.0	59894.0	70437.0	77271.0	82068.0
52.0	-117055.0	31883.0	37630.0	45469.0	51013.0	55193.0
64.0	-114283.0	19575.0	23328.0	28678.0	32647.0	35736.0
76.0	-103125.0	11144.0	13447.0	16279.0	19258.0	21248.0
90.0	-49901.0	3047.0	4772.0	6004.0	7092.0	7894.0
105.0	-7353.0	-2882.0	-3241.0	-3822.0	-4330.0	-4768.0

EE(J,N) 0.0 0.0 0.0 0.0 0.0 0.0

WING/PYLON DATA

NSW = 8	NSHP = 4	WZETA = 0.010	PM = 7.90 CHUG	ZP = 81.80 IN	YP = 0.0 IN
XP = 0.0 IN	NPOD = 5	HM = 0.01 CHUG	WEPE = 9.35 IN	WIPP = 16390. CHUG-IN**2	
SHPE(1) = -1.100	SHPE(2) = 0.258	SHPE(3) = -0.730	SHPE(4) = -0.370		

SEGMENT(I)	1	2	3	4	5	6	7	8
PHX(I,J) (IN/IN)	0.01020	0.01070	0.01480	0.01720	0.02090	0.02570	0.02650	0.03390
	0.31870	0.29400	0.17030	0.07890	-0.05170	-0.29733	-0.36690	-0.79630
	-0.02040	-0.02080	-0.02710	-0.03180	-0.03960	-0.05740	-0.06370	-0.10510
	-0.00900	-0.00620	0.00180	0.00460	0.00440	-0.00740	-0.01340	-0.07040
PHZ(I,J) (IN/IN)	0.36930	0.34270	0.21100	0.11090	-0.03350	-0.31710	-0.39750	-0.88100
	0.02550	0.02270	0.01080	0.00140	-0.01190	-0.03600	-0.04260	-0.07710
	-0.09700	-0.06600	0.06350	0.14850	0.25440	0.40720	0.44040	0.56260
	-0.29240	-0.31160	-0.35070	-0.33890	-0.27720	-0.06790	0.00910	0.49160
PHT(I,J) (RAD/IN)	0.0	0.0	0.00010	0.00030	0.00070	0.00100	0.00140	0.00170
	0.0	-0.00010	-0.00030	-0.00050	-0.00070	-0.00090	-0.00110	-0.00120
	0.0	0.00140	0.00270	0.00400	0.00530	0.00670	0.00800	0.00930
	0.00030	0.00140	0.00250	0.00360	0.00470	0.00580	0.00690	0.00800
WM(I) (CHUG)	20.0434	8.1100	6.5500	8.1800	4.8900	7.3800	3.5700	0.8010
WIE(I) (CHUG-IN**2)	0.0	0.0	0.0	15760.0000	0.0	15760.0000	0.0	0.0
WY(I) (IN)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
WDR(I) (IN)	0.0	37.50	55.00	37.50	50.00	37.50	47.50	40.00

J	OM(J) RAD/SEC	PHXA(J) IN/IN	PHYA(J) IN/IN	PHZA(J) IN/IN	PHPA(J) RAD/IN	PHTA(J) RAD/IN	FM(J) CHUG
1	17.480	0.03610	0.0	-1.00000	-0.00026	0.00380	13.2503
2	33.060	-1.00000	0.0	-0.13230	0.00650	-0.00140	16.2371
3	40.690	-0.13160	0.0	1.00000	0.00010	0.01510	9.8204
4	60.640	-0.11000	0.0	1.00000	0.00077	0.00980	3.0095

ARAP06 Sample Problem, Input Display (Continued)

ACCELEROMETER LOCATIONS

K	1	2	3
PX(K) IN	0.0	0.0	0.0
PY(K) IN	0.0	0.0	0.0
PZ(K) IN	0.0	83.000	103.000

BELL HELICOPTER COMPANY

REAL TIME LIMIT = 1.00 SEC

END OF INPUT DATA

ARAP06 Sample Problem, Input Display (Continued)

TIME FI 1BM1 ZTIP1 THF XPDD1	RPM FJ 1CM1 XTIP1 PSIP YPDD1	PSI FK 2BM1 ZTIP2 XHB ZPDD1	BET-LG MI 2CM1 XTIP2 YHB XPDD2	BET-LT MJ 3BM1 ZTIP3 ZHB YPDD2	BET(1) MK 3CM1 XTIP3 WBM ZPDD2	FFX TFFX 1BM2 1QB1 WCD XPDD3	FFY TFFY 1CM2 2QB1 WTN YPDD3	FFZ TFFZ 2BM2 3QB1 QW1 ZPDD3	WBX TFMX 2CM2 1QB2 QW2 XPDD4	WBY TFMY 3BM2 2QB2 QW3 YPDD4	WBZ TFMZ 3CM2 3QB2 QW4 ZPDD4
0.004	160.	4.3	0.001	-0.003	3.000	-317.6	152.7	159.5	-3083.2	0.0	0.0
527.	-103.	-2068.	71764.	17730.	272606.	-2964.0	130.0	-1908.3	-88.0	5945.2	2853.8
3705.	-456.	3705.	-979.	3705.	1435.	953.	-1509.	761.	-1655.	1646.	-979.
0.064	0.013	0.061	0.001	0.074	0.055	0.033	0.033	0.033	-0.005	-0.011	0.016
-0.019	0.002	-0.183	0.195	2.290	0.364	-0.140	-0.012	-1.095	0.256	-0.721	-0.359
-1.252	0.000	-0.004	0.491	-0.026	-0.004	0.891	-0.032	-0.004			
0.009	160.	8.6	0.018	-0.011	3.015	-792.8	98.3	-143.0	-3082.6	-0.1	58.3
-2557.	923.	-1640.	152037.	-6983.	195493.	-6574.4	424.2	-1724.4	-3137.1	15111.9	5622.7
9256.	-1791.	9256.	-4405.	9256.	6196.	2142.	-3954.	1185.	-4687.	5067.	-1713.
0.125	0.011	0.113	-0.041	0.161	0.167	0.083	0.083	0.083	-0.020	-0.049	0.058
-0.019	0.002	-0.183	0.194	2.277	0.376	-0.140	-0.012	-1.094	0.253	-0.711	-0.344
-1.617	-0.000	0.038	-0.002	-0.040	0.037	0.368	-0.049	0.037			
0.013	160.	12.9	0.038	-0.020	3.030	-496.8	11.9	-473.3	-3082.6	-0.1	57.6
-1304.	1009.	-1240.	95822.	4394.	130913.	-5124.7	598.7	-1656.2	-5014.4	8258.1	7881.3
16862.	-3690.	16862.	-8863.	16862.	12553.	3745.	-7323.	1852.	-8774.	9695.	-2765.
0.205	0.014	0.185	-0.071	0.268	0.285	0.151	0.151	0.151	-0.041	-0.098	0.139
-0.018	0.002	-0.181	0.193	2.255	0.393	-0.139	-0.011	-1.091	0.249	-0.695	-0.322
-1.641	-0.000	0.047	0.055	-0.049	0.044	0.443	-0.060	0.044			
0.019	160.	17.3	0.069	-0.031	3.053	-127.0	-57.2	-839.7	-3082.7	-0.1	55.9
-1824.	1505.	-708.	10092.	27831.	55964.	-5459.5	671.2	-1491.8	-5529.8	2822.8	8652.7
26088.	-6193.	26088.	-14520.	26089.	20712.	5619.	-11465.	2569.	-13802.	15472.	-3919.
0.294	0.013	0.264	-0.114	0.391	0.428	0.234	0.234	0.234	-0.058	-0.160	0.229
-0.018	0.002	-0.180	0.191	2.228	0.416	-0.138	-0.011	-1.098	0.243	-0.674	-0.295
-1.366	-0.000	0.076	0.028	-0.063	0.072	0.352	-0.078	0.071			
0.022	160.	21.6	0.108	-0.039	3.080	313.1	-89.9	-1254.6	-3082.7	-0.1	54.1
-1771.	1729.	-170.	-79623.	59536.	-31654.	-5118.2	691.7	-1370.3	-5570.2	-4577.5	9375.3
36022.	-8914.	36022.	-20595.	36022.	29509.	7625.	-15934.	3348.	-19210.	21697.	-5155.
0.332	0.014	0.343	-0.152	0.511	0.566	0.323	0.323	0.323	-0.098	-0.228	0.326
-0.017	0.002	-0.177	0.188	2.196	0.442	-0.137	-0.010	-1.085	0.235	-0.650	-0.265
-1.021	-0.000	0.107	0.095	-0.081	0.101	0.346	-0.099	0.100			
0.027	160.	25.9	0.154	-0.044	3.111	783.3	-69.2	-1676.1	-3082.7	-0.1	52.0
-1944.	1947.	344.	-172260.	95496.	-125687.	-4958.6	635.2	-1280.2	-4302.3	-12324.9	7760.8
45622.	-11412.	45622.	-26262.	45622.	37674.	9613.	-20214.	4174.	-24380.	27589.	-6444.
0.459	0.017	0.413	-0.182	0.613	0.678	0.409	0.409	0.409	-0.126	-0.290	0.416
-0.016	0.002	-0.173	0.184	2.161	0.470	-0.134	-0.010	-1.081	0.226	-0.622	-0.234
-0.542	-0.000	0.143	0.168	-0.099	0.137	0.331	-0.122	0.135			
0.031	160.	30.2	0.202	-0.042	3.143	1200.4	11.6	-2018.4	-3082.8	-0.1	49.7
-1993.	1987.	705.	-245701.	126290.	-204751.	-4655.0	515.9	-1264.2	-397.3	-17986.4	6636.9
53949.	-13214.	53949.	-30681.	53949.	43896.	11470.	-23825.	5073.	-28725.	32365.	-7804.
0.519	0.027	0.468	-0.195	0.687	0.747	0.483	0.483	0.483	-0.146	-0.339	0.485
-0.015	0.001	-0.168	0.178	2.122	0.498	-0.131	-0.009	-1.076	0.215	-0.592	-0.204
-0.081	-0.000	0.175	0.270	-0.116	0.168	0.351	-0.142	0.167			

ARAP06 Sample Problem, Condensed Output

ROTOR AIRLOADS DETAILS

STAT	BLADE	VP DHF	VT DQ	PHI DRADF	AL DTHST	CL DLIFT	CD DDRAG	CM DMOMT	MACH
1	1	-3547. -348.	497. 15.	-82.02 -8.20	-21.48 -137.55	-0.674 -364.	0.200 88.	-0.001 -19.	0.267
2	1	-3535. -631.	998. -939.	-74.23 -16.76	-16.20 -281.07	-0.736 -684.	0.107 100.	-0.026 -973.	0.274
3	1	-3523. -530.	1459. -1881.	-66.97 -15.62	-11.43 -262.02	-0.586 -591.	0.034 34.	-0.047 -1877.	0.286
4	1	-3513. -418.	1956. -2606.	-60.40 -15.37	-7.36 -257.80	-0.437 -491.	0.016 18.	-0.057 -2570.	0.301
5	1	-3505. -410.	2492. -499.	-54.59 -18.27	-4.05 -306.32	-0.402 -512.	0.010 12.	-0.009 -435.	0.321
6	1	-3499. -116.	2986. -551.	-49.52 -6.67	-1.48 -111.79	-0.110 -161.	0.007 10.	-0.008 -466.	0.343
7	1	-3494. 162.	3477. -1283.	-45.14 8.78	0.41 147.31	0.131 219.	0.006 10.	-0.017 -1164.	0.368
8	1	-3491. 371.	3967. -1521.	-41.35 24.04	1.71 403.13	0.285 548.	0.006 12.	-0.019 -1434.	0.394
9	1	-3489. 539.	4455. -1962.	-38.06 39.72	2.49 666.14	0.389 858.	0.006 13.	-0.021 -1879.	0.422
10	1	-3487. 326.	4944. -1127.	-35.20 26.69	2.86 447.59	0.440 554.	0.006 8.	-0.022 -1087.	0.451
1	2	-3569. -341.	473. 79.	-82.46 -5.14	-22.49 -143.91	-0.300 -357.	0.220 98.	0.005 98.	0.269
2	2	-3572. -671.	976. -834.	-74.72 -10.90	-17.26 -304.98	-0.771 -727.	0.125 118.	-0.022 -819.	0.276
3	2	-3576. -570.	1479. -1719.	-67.53 -10.31	-12.56 -298.27	-0.618 -637.	0.047 49.	-0.042 -1718.	0.289
4	2	-3579. -500.	1983. -2610.	-61.01 -10.77	-8.55 -301.29	-0.506 -583.	0.019 21.	-0.057 -2614.	0.305
5	2	-3582. -578.	2487. -457.	-55.23 -14.93	-5.27 -417.54	-0.545 -713.	0.010 14.	-0.009 -462.	0.325
6	2	-3586. -280.	2991. -486.	-50.17 -3.88	-2.70 -248.43	-0.249 -374.	0.008 12.	-0.008 -488.	0.348
7	2	-3589. 14.	3495. -1178.	-45.76 -0.10	-0.80 -2.77	0.005 9.	0.007 12.	-0.017 -1174.	0.374
8	2	-3593. 209.	3999. -1405.	-41.94 7.69	0.53 215.20	0.151 300.	0.006 12.	-0.018 -1394.	0.401
9	2	-3597. 384.	4503. -1984.	-38.61 16.41	1.35 453.08	0.262 599.	0.006 14.	-0.022 -1973.	0.430
10	2	-3601. 248.	5007. -1168.	-35.72 11.93	1.74 330.99	0.316 414.	0.006 8.	-0.022 -1161.	0.460

ARAP06 Sample Problem, Condensed Output (Continued)

1	3	-3317.	224.	-86.14	-25.44	-0.729	0.280	0.023	0.248
		-270.	369.	-7.71	-124.56	-277.	106.	345.	
2	3	-3333.	788.	-76.70	-18.50	-0.811	0.145	-0.016	0.256
		-610.	-377.	-16.34	-264.19	-655.	117.	-529.	
3	3	-3358.	1336.	-68.31	-12.61	-0.620	0.048	-0.041	0.270
		-502.	-1489.	-15.19	-245.54	-557.	43.	-1492.	
4	3	-3386.	1871.	-61.07	-7.88	-0.480	0.017	-0.058	0.289
		-424.	-2432.	-15.70	-253.73	-494.	17.	-2388.	
5	3	-3414.	2398.	-54.92	-4.23	-0.423	0.010	-0.009	0.311
		-437.	-465.	-18.57	-300.18	-506.	12.	-414.	
6	3	-3442.	2917.	-49.72	-1.53	-0.115	0.007	-0.008	0.337
		-117.	-472.	-6.92	-111.87	-162.	10.	-448.	
7	3	-3468.	3432.	-45.30	0.38	0.127	0.006	-0.017	0.364
		155.	-1122.	8.62	139.41	209.	10.	-1139.	
8	3	-3493.	3942.	-41.54	1.64	0.278	0.006	-0.019	0.393
		360.	-1378.	24.05	388.80	531.	11.	-1420.	
9	3	-3518.	4450.	-38.32	2.36	0.376	0.006	-0.021	0.423
		527.	-1830.	39.92	643.75	833.	13.	-1892.	
10	3	-3541.	4956.	-35.54	2.64	0.417	0.006	-0.022	0.454
		316.	-1071.	26.49	428.19	533.	8.	-1109.	

WING AIRLOADS DETAILS

STAT	VP	VT	PH	AL	CL	CD	WL	WD	WT	WM	WQ
1	4.62	3548.	0.07	9.07	0.915	0.015	0.	0.	0.	0.	0.
2	4.56	3548.	0.07	9.03	0.911	0.015	3298.	55.	3288.	51.	30446.
3	3.87	3548.	0.06	8.97	0.905	0.015	4793.	80.	4793.	75.	44381.
4	3.04	3547.	0.05	8.92	0.899	0.015	3246.	54.	3246.	51.	30064.
5	1.54	3547.	0.02	8.84	0.892	0.015	4293.	72.	4293.	70.	39762.
6	-2.02	3546.	-0.03	8.74	0.882	0.015	3130.	53.	3180.	55.	29463.
7	-3.11	3546.	-0.05	8.67	0.875	0.015	3998.	67.	3998.	70.	37045.
8	-5.95	3545.	-0.16	8.52	0.859	0.014	3304.	55.	3304.	64.	30632.

ARAP06 Sample Problem, Condensed Output (Continued)

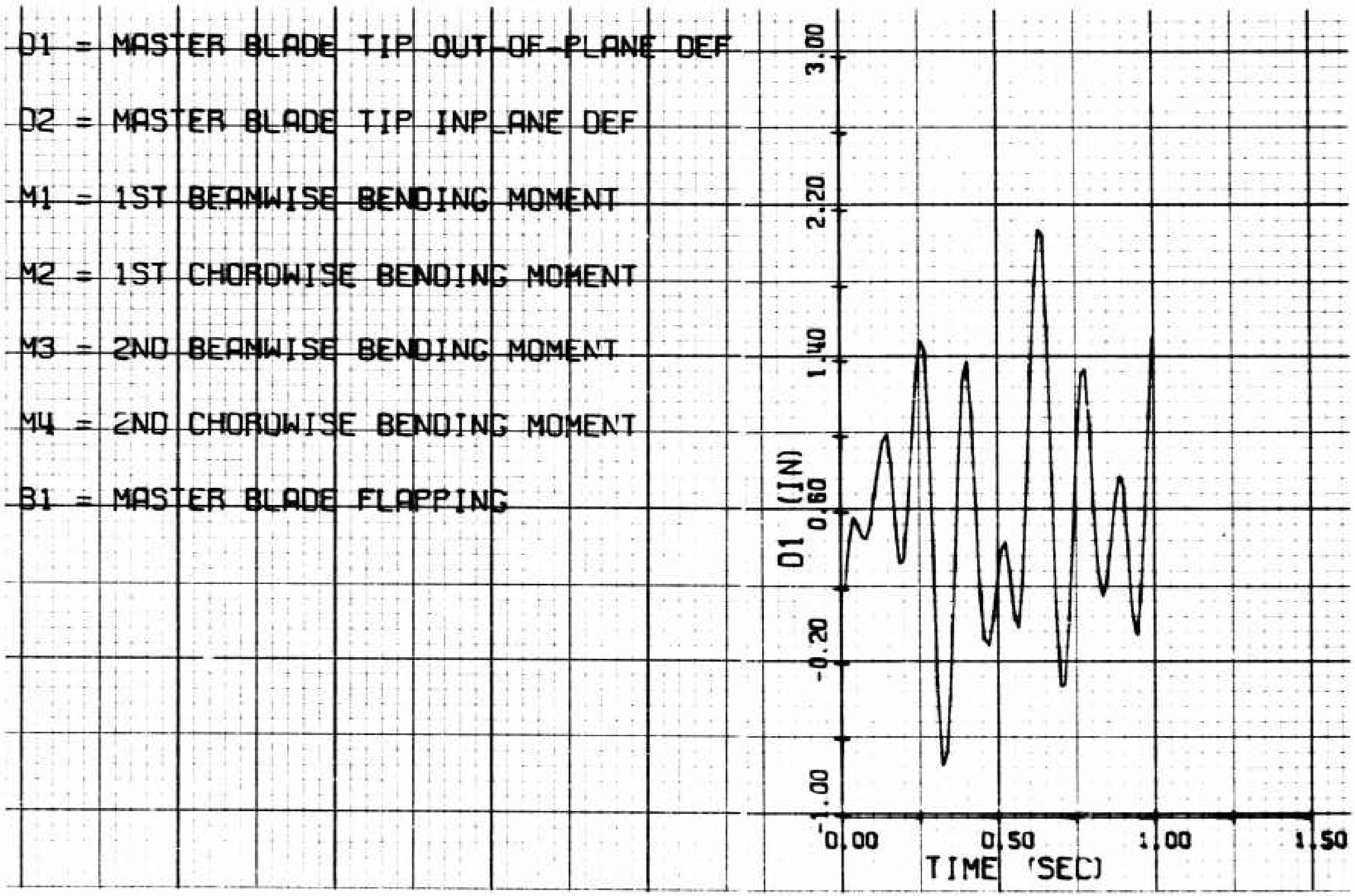
TIME	BET(1)	TIME	BET(1)	TIME	BET(1)	TIME	BET(1)	TIME	BET(1)	TIME	BET(1)
0.063	3.267	0.157	2.881	0.175	2.832	0.180	2.892	0.184	2.894	0.189	2.894
0.288	3.719	0.432	1.664	0.661	4.258	0.859	2.145				

TIME	WBM	TIME	WBM	TIME	WBM	TIME	WBM	TIME	WBM	TIME	WBM
0.058	0.586	0.202	-0.760	0.225	-0.747	0.283	-0.862	0.495	0.314	0.517	0.302
0.554	0.595	0.679	0.191	0.702	0.202	0.805	-0.359	0.832	-0.338	0.909	-0.631
0.981	-0.337										

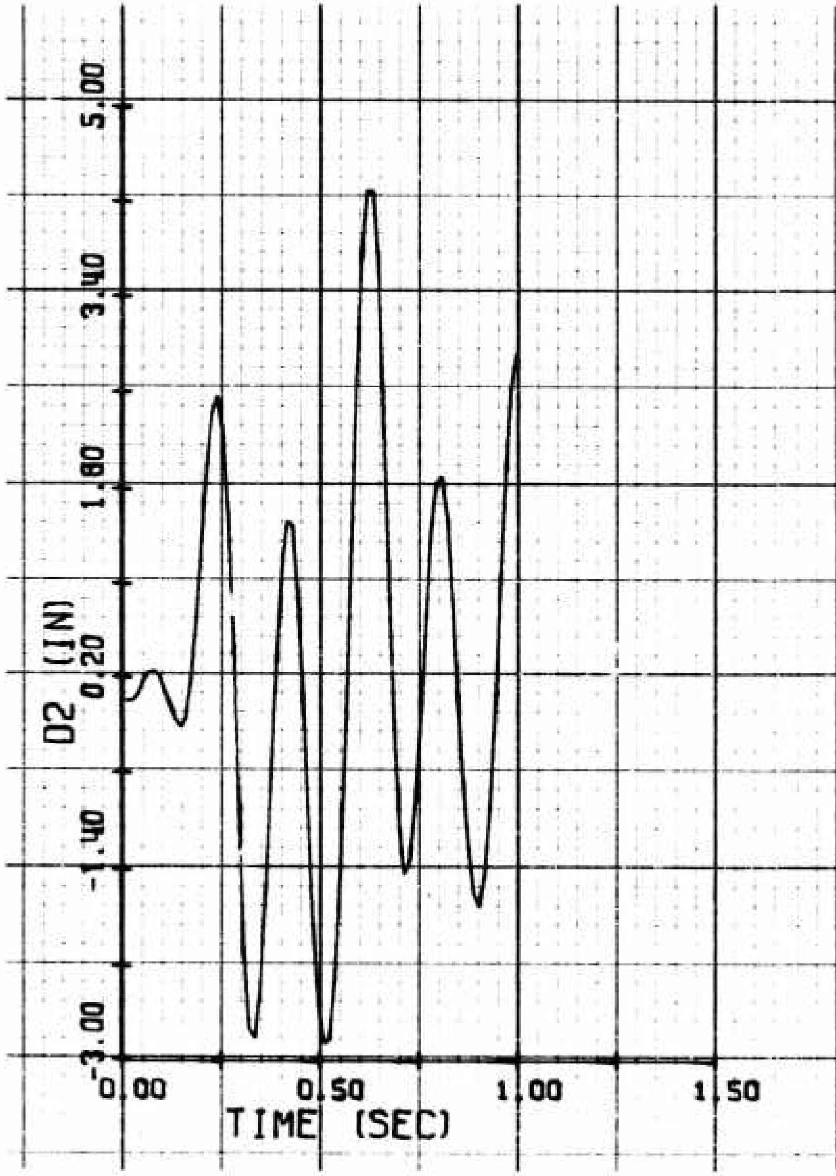
TIME	WCD	TIME	WCD	TIME	WCD	TIME	WCD	TIME	WCD	TIME	WCD
0.004	-0.140	0.112	-0.018	0.135	-0.021	0.189	0.005	0.261	-0.127	0.333	0.047
0.400	-0.089	0.450	-0.031	0.517	-0.153	0.595	-0.007	0.643	-0.134	0.706	0.094
0.769	-0.102	0.832	0.028	0.895	-0.086	0.954	0.020				

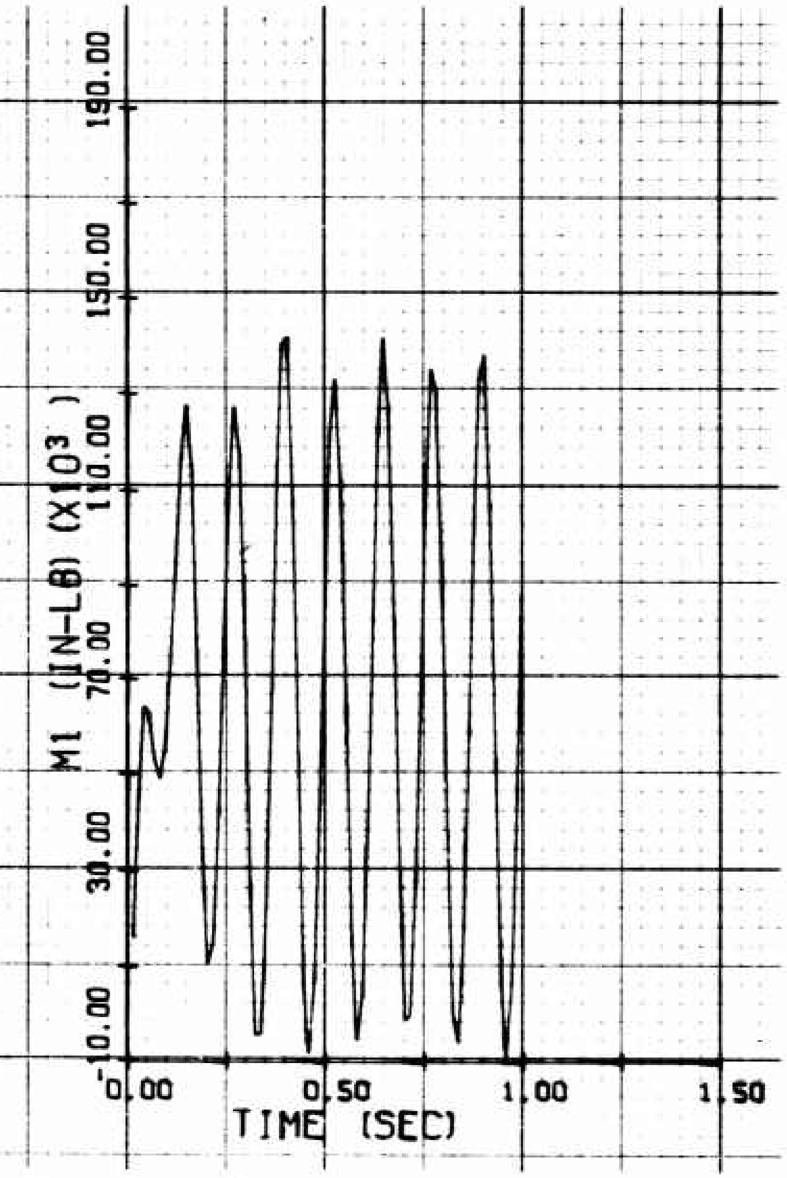
TIME	SHP(1)	TIME	SHP(1)	TIME	SHP(1)	TIME	SHP(1)	TIME	SHP(1)	TIME	SHP(1)
0.004	-1.095	0.252	0.664	0.576	-1.019	0.891	0.285				

ARAP06 Sample Problem, Condensed Output (Concluded)

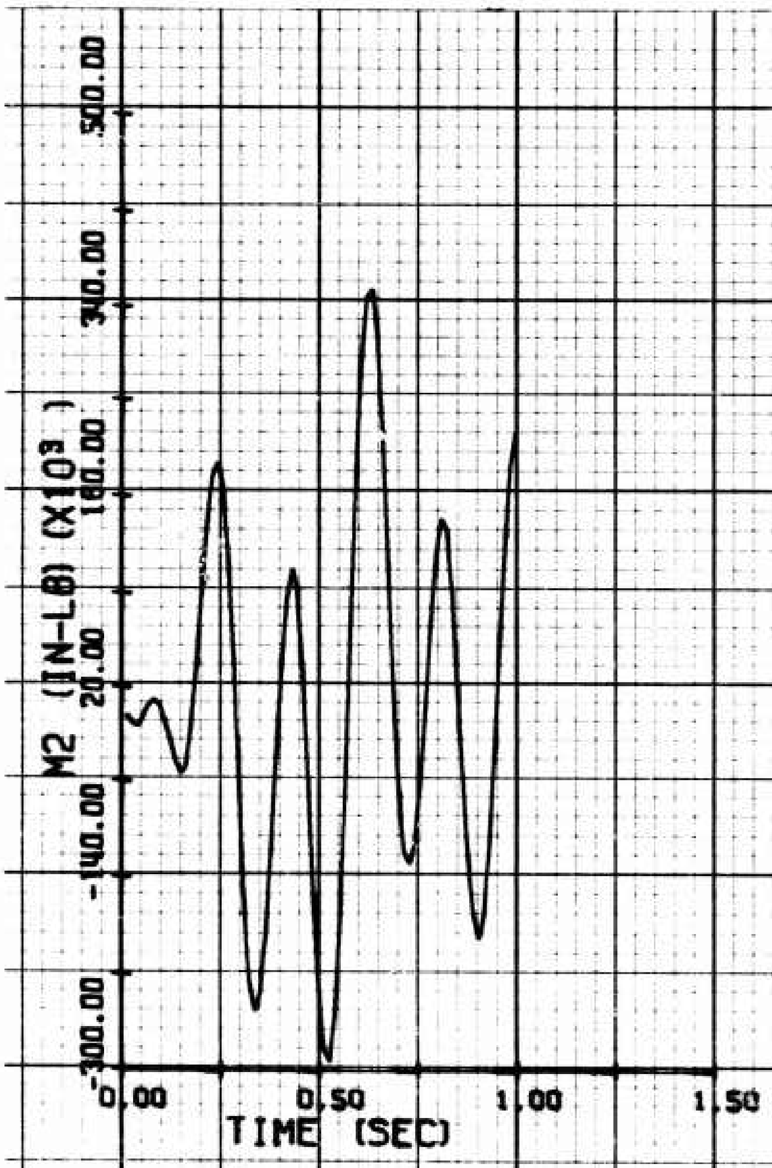


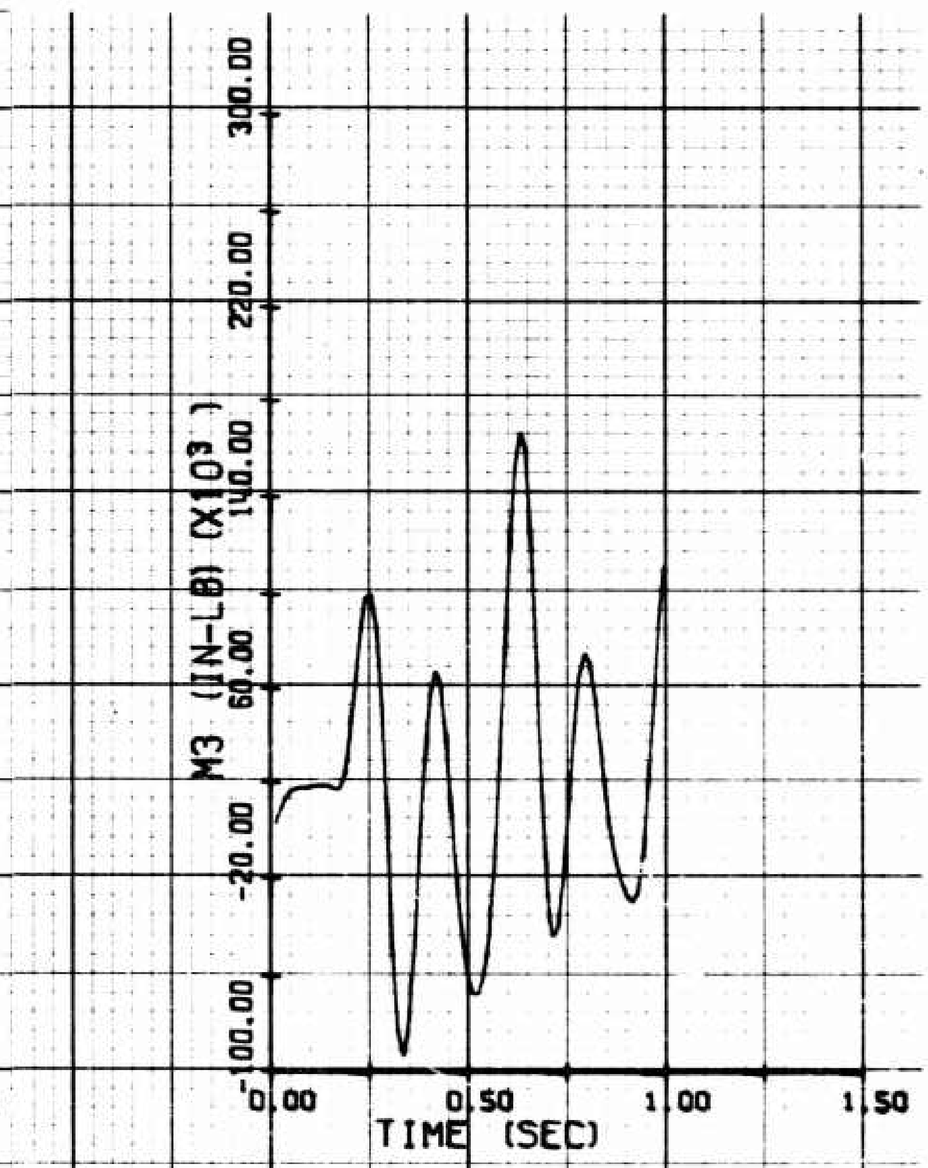
ARAF06 Sample Problem, CALCOMP Plots



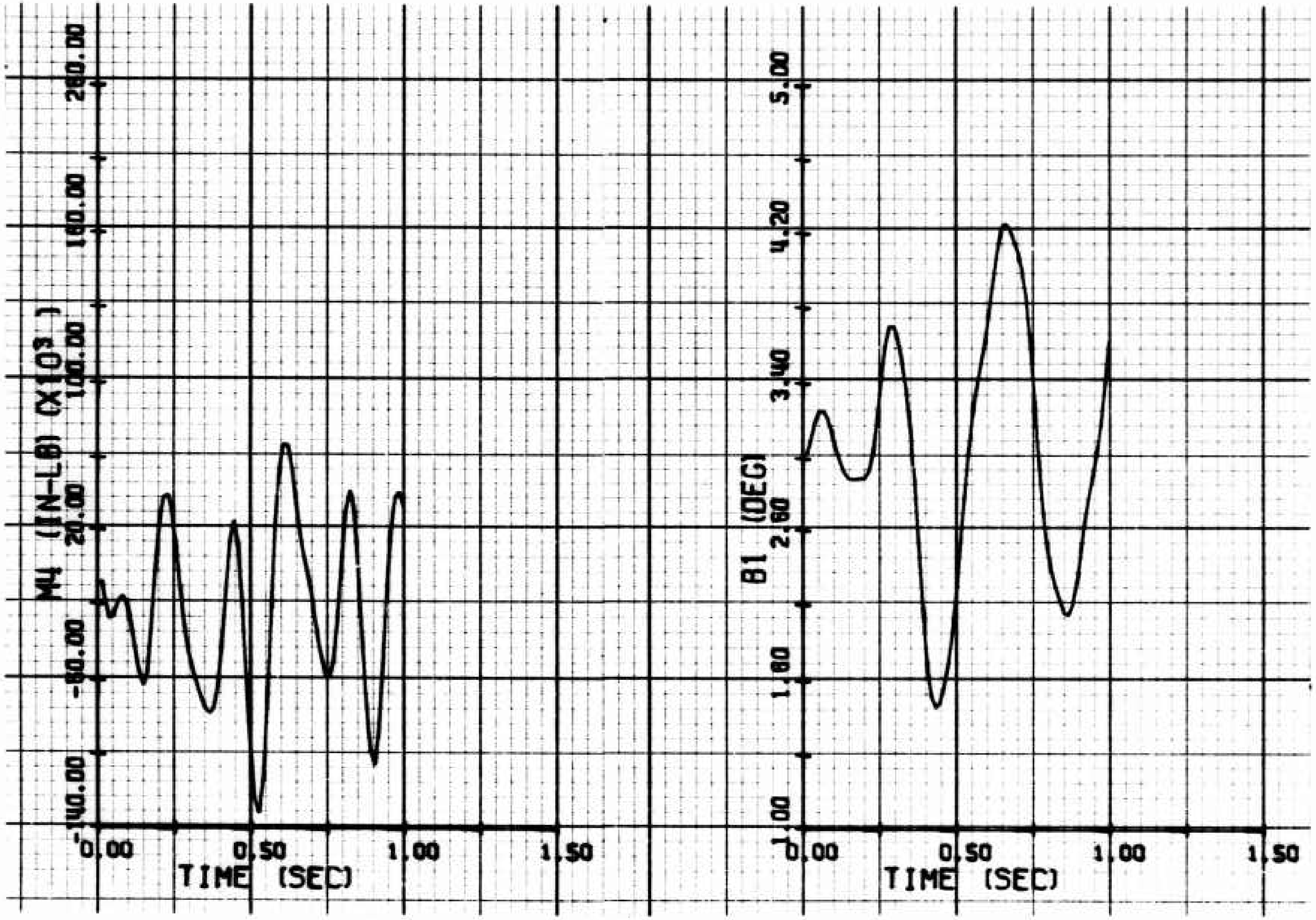


ARAP06 Sample Problem, CALCOMP Plots (Continued)





ARAP06 Sample Problem, CALCOMP Plots (Continued)



ARAP06 Sample Problem, CALCOMP Plots (Concluded)

DATA SHEET

D270 TRANSITION MODE FLUTTER ANALYSIS						CARD 01
-.74 6 15		1.E-6				CARD 02
4.88	97.075	4.88	0.	1.	3.	CARD 03
3.5	4.	5.				
.99452	4.88	97.075	603.	11.2		CARD 04
4.15	2.73	2.43	2.6	2.71	0.	CARD 05
0.						
5.75	5.426	5.16	4.88	4.59	1.	CARD 06
1.						
5.6	5.6	5.6	5.6	5.6	1.	CARD 07
1.						
.86603	4.88	97.075	603.	11.2		CARD 08
0.	3.93	4.42	3.33	2.5		CARD 09
1.52	1.9	2.	2.	1.75		CARD 10

DFAL17 Sample Problem, Input Data

1.667		1.667		1.667		1.667		1.667			CARD 11
.86603		4.88		97.075		603.		11.2			CARD 12
0.		3.93		4.42		3.33		2.5			CARD 13
1.52		1.9		2.		2.		1.75			CARD 14
1.667		1.667		1.667		1.667		1.667			CARD 15
.86603		4.88		97.075		603.		11.2			CARD 16
0.		3.93		4.42		3.33		2.5			CARD 17
1.52		1.9		2.		2.		1.75			CARD 18
1.667		1.667		1.667		1.667		1.667			CARD 19
-7.5		30.			1.E+6		1.E+6		1.E+6	10.	CARD 20
	1.E+6		1.E+6	29.4		59.		122.		27.	CARD 21
55.		46000.		180000.		22000.					CARD 22
55.		34000.		155000.		18000.					CARD 23

DATA SHEET

DFAL17 Sample Problem, Input Data (Continued)

46.	25000.	125000.	14500.			CARD 24
46.	17500.	100000.	12500.			CARD 25
50.	16000.	90000.	10500.			CARD 26
3.	1500.	1500.	100000.			CARD 27
60.	1800.	12500.	100000.			CARD 28
60.	750.	12000.	100000.			CARD 29
75.	300.	9500.	100000.			CARD 30
55.5	110.	8450.	100000.			CARD 31
13.4	13.4	13.4	13.4	13.4	122.0	CARD 32
10.	.1	.1	.1	.1	.1	CARD 33
53.6	53.6	53.6	53.6	53.6	10.	CARD 34
10.	20.0	20.0	20.0	20.0	20.0	CARD 35
1730.	3370.	1450.	2950.	1450.	4187.	CARD 36
4187	170.	213.2	105.7	108.5	46.	CARD 37

DATA SHEET

DFAL17 Sample Problem, Input Data (Continued)

12800.	43700.	9650.	38050.	9650.	-39000.	CARD 38
10.	.1	.1	.1	.1	.1	CARD 39
3300000.	7540000.	1660000.	6250000.	120000.	44800000.	CARD 40
4480000.	100.	42200.	8700.	12500.	9650.	CARD 41

DATA SHEET

DFAL17 Sample Problem, Input Data (Concluded)

FOLD-UNFOLD ROTOR VIBRATION AND FLUTTER ANALYSIS BY A COLLOCATION METHOD USING AERODYNAMIC INFLUENCE COEFFICIENTS

D270 TRANSITION MODE FLUTTER ANALYSIS

CONTROL DATA

```
NSIZE              -74
NAERO                4
MODES OUT           15
FLEX K     0.10000E-05
MASS K     0.10000E 01
DENSITY    0.23780E-02
BR         0.48800E 01
S          0.97075E 02
BIREFI     0.48800E 01
```

REDUCED VELOCITIES

```
0.0          0.10000E 01    0.30000E 01    0.35000E 01    0.40000E 01    0.50000E 01
```

AERODYNAMIC INPUT DATA

SURFACE	COSINE LAMBDA	BR	S	CAP S	C BAR
1	0.99452E 00	0.48800E 01	0.97075E 02	0.60300E 03	0.11200E 02
2	0.46603E 00	0.48800E 01	0.97075E 02	0.60300E 03	0.11200E 02
3	0.86603E 00	0.48800E 01	0.97075E 02	0.60300E 03	0.11200E 02
4	0.86603E 00	0.48800E 01	0.97075E 02	0.60300E 03	0.11200E 02

STRIP		1	2	3	4	5	6	7
SURFACE	DY	0.41500E 01	0.27300E 01	0.24300E 01	0.26300E 01	0.27100E 01	0.0	0.0
1	B	0.57530E 01	0.54260E 01	0.51600E 01	0.48800E 01	0.45900E 01	0.10000E 01	0.10000E 01
	D	0.56000E 01	0.56000E 01	0.56000E 01	0.56000E 01	0.56000E 01	0.10000E 01	0.10000E 01
SURFACE	DY	0.0	0.39300E 01	0.44200E 01	0.33300E 01	0.25000E 01		
2	B	0.15200E 01	0.19000E 01	0.20000E 01	0.20000E 01	0.17500E 01		
	D	0.16670E 01	0.16670E 01	0.16670E 01	0.16670E 01	0.16670E 01		
SURFACE	DY	0.0	0.39300E 01	0.44200E 01	0.33300E 01	0.25000E 01		
3	B	0.15200E 01	0.19000E 01	0.20000E 01	0.20000E 01	0.17500E 01		
	D	0.16670E 01	0.16670E 01	0.16670E 01	0.16570E 01	0.16670E 01		
SURFACE	DY	0.0	0.39300E 01	0.44200E 01	0.33300E 01	0.25000E 01		
4	B	0.15200E 01	0.19000E 01	0.20000E 01	0.20000E 01	0.17500E 01		
	D	0.16670E 01	0.16670E 01	0.16670E 01	0.16670E 01	0.16670E 01		

DFAL17 Sample Problem, Input Display

FLEXIBILITY MATRIX DATA

WING SWEEP	-7.50000 (DEGREES)
BLADE SWEEP	30.00000 (DEGREES)
DIST-EA TO CONVERSION AXIS	29.400 (INCHES)
PYLON OFFSET	59.00000 (INCHES)
MAST LENGTH	122.00000 (INCHES)
HUB OFFSET	27.00000 (INCHES)
PYLON-F/A STIFFNESS	1000000.0 (IN-LBS/RADIAN)
PYLON-LAT STIFFNESS	1000000.0 (IN-LBS/RADIAN)
BLADE FOLD STIFFNESS	1000000.0 (IN-LBS/RADIAN)
FIXED CONTROL STIFFNESS	1000000.0 (IN-LBS/RADIAN)
MOVABLE CONTROL STIFFNESS	1000000.0 (IN-LBS/RADIAN)
MAST STIFFNESS	10.0 (IN-LBS/RADIAN)
BEAMWISE STIFFNESS(CONVERSION AXIS)	2324.4 (LBS/IN)
CHORDWISE STIFFNESS(CONVERSION AXIS)	11515.2 (LBS/IN)

WING PARAMETERS

SEG	LENGTH	EIB	EIC	GJ	X	Y	M	S	I
1	0.55000E 02	0.46000E 05	0.18000E 06	0.22000E 05	0.134000D 02	0.536000D 02	0.193000D 04	0.128000D 05	0.333330D 07
2	0.55000E 02	0.34000E 05	0.15500E 06	0.18000E 05	0.134000D 02	0.536000D 02	0.337000D 04	0.437000D 05	0.754330D 07
3	0.46000E 02	0.25000E 05	0.12500E 06	0.14500E 05	0.134000D 02	0.536000D 02	0.145000D 04	0.965000D 04	0.166000D 07
4	0.46000E 02	0.17500E 05	0.10000E 06	0.12500E 05	0.134000D 02	0.536000D 02	0.295000D 04	0.383500D 05	0.575000D 07
5	0.50000E 02	0.16000E 05	0.90000E 05	0.13500E 05	0.134000D 02	0.536000D 02	0.145000D 04	0.965000D 04	0.123330D 06

PYLON PARAMETERS

	X	Y	M	S	I
PITCH	0.1220000 03	0.1000000 02	0.4187000 04	-0.3930000 05	0.4480000 08
YAW	0.1000000 02	0.1000000 02	0.4187000 04	0.1000000 02	0.4480000 08

BLADE PARAMETERS

SEG	LENGTH	EIB	EIC	GJ	X	Y	M	S	I
1	0.30000E 01	0.15000E 04	0.15000E 04	0.10000E 06	0.100000D 00	0.200000D 02	0.170000D 03	0.100000D 00	0.100000D 03
2	0.60000E 02	0.18000E 04	0.12500E 05	0.10000E 06	0.100000D 00	0.200000D 02	0.213700D 03	0.100000D 00	0.422330D 05
3	0.60000E 02	0.75000E 03	0.12000E 05	0.10000E 06	0.100000D 00	0.200000D 02	0.105700D 03	0.100000D 00	0.873330D 06
4	0.75000E 02	0.30000E 03	0.95000E 04	0.10000E 06	0.100000D 00	0.200000D 02	0.108500D 03	0.100000D 00	0.125000D 05
5	0.55500E 02	0.11000E 03	0.84500E 04	0.10000E 06	0.100000D 00	0.200000D 02	0.460000D 02	0.100000D 00	0.965000D 06

DFAL17 Sample Problem, Input Display (Continued)

FLEXIBILITY MATRIX

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	[illegible]	-0.2627360E 00	0.3513446E 01	[illegible]	0.5541671E 01	0.2859077E 01
2	[illegible]	0.7459778E 01	0.5518420E 01	[illegible]	0.1542040E 01	0.1041050E 02
3	[illegible]	0.2317209E 00	0.1179947E 02	0.5417598E 01	0.1922292E 02	0.1243186E 02
4	[illegible]	0.3236167E 01	0.5417514E 01	0.2339114E 02	0.1114464E 02	0.2950539E 02
5	[illegible]	0.1632146E 01	0.1322930E 02	0.1114859E 02	0.3442276E 02	0.2412411E 02
6	[illegible]	0.1031053E 02	0.1243371E 02	0.2950542E 02	0.2312785E 02	0.5111417E 02
7	[illegible]	0.2675501E 01	0.2660970E 02	0.1689885E 02	0.5379379E 02	0.3665372E 02
8	[illegible]	0.1140154E 02	0.1954906E 02	0.3532500E 02	0.3442141E 02	0.6429448E 02
9	[illegible]	0.3835016E 01	0.3477423E 02	0.2316843E 02	0.6944093E 02	0.5192711E 02
10	0.6441282E 01	0.1257508E 02	0.2743669E 02	0.4155676E 02	0.5559115E 02	0.7868893E 02
11	[illegible]	-0.3541070E 01	0.5642014E 02	0.1418966E 02	0.1122973E 03	0.4420020E 02
12	0.1001679E 02	0.1357060E 02	0.4187753E 02	0.5930129E 02	0.8659819E 02	0.1062276E 03
13	0.5495735E-01	-0.2292163E-01	0.1926419E 00	0.2135322E-01	0.3755012E 00	0.1093197E 00
14	[illegible]	-0.2543180E-01	0.2142453E 00	0.2372845E-01	0.4175088E 00	0.1215172E 00
15	0.3112240E-01	-0.1383161E-01	0.1160945E 00	0.1280491E-01	0.2261816E 00	0.6571722E-01
16	0.2740540E-01	-0.2812161E-01	0.2363091E 00	0.2615879E-01	0.4594912E 00	0.1339789E 00
17	[illegible]	-0.1957417E 01	0.5978493E 02	0.1955080E 02	0.1196605E 03	0.5736764E 02
18	0.1543451E 02	0.6695329E 00	0.5753908E 02	0.2506178E 02	0.1157473E 03	0.6532022E 02
19	0.2369338E-01	-0.9976346E-02	0.8374918E-01	0.9242032E-02	0.1631698E 00	0.4761926E-01
20	[illegible]	-0.3479913E-01	0.2923327E 00	0.3232989E-01	0.5596307E 00	0.1656919E 00
21	0.1618905E 02	0.4309176E 01	0.6218280E 02	0.3603119E 02	0.1270100E 03	0.8639255E 02
22	0.1532559E 02	0.6936128E 01	0.5993700E 02	0.4154292E 02	0.1230967E 03	0.9466509E 02
23	[illegible]	-0.6121162E-02	0.5140659E-01	0.5679198E-02	0.1001593E 00	0.2912153E-01
24	0.9937537E-01	-0.4147650E-01	0.3483553E 00	0.3850090E-01	0.6787682E 00	0.1973844E 00
25	0.1608037E 02	0.1057577E 02	0.6458070E 02	0.5251305E 02	0.1343594E 03	0.1157374E 03
26	0.1521698E 02	0.1320272E 02	0.6233600E 02	0.5402405E 02	0.1334461E 03	0.1240080E 03
27	0.3128613E-02	-0.1302252E-02	0.1097430E-01	0.1225696E-02	0.2139716E-01	0.6249756E-02
28	0.1193562E 00	-0.4982311E-01	0.4183826E 00	0.4621460E-01	0.8151883E 00	0.2369996E 00
29	[illegible]	0.1346901E 02	0.6757800E 02	0.7311446E 02	0.1635446E 03	0.1524185E 03
30	0.1508078E 02	0.2103596E 02	0.6533220E 02	0.7467546E 02	0.1396329E 03	0.1606910E 03
31	-0.5407955E-02	0.2263736E-02	-0.1894407E-01	-0.2069901E-02	-0.3688678E-01	-0.1067539E-01
32	[illegible]	-0.5599958E-01	0.4702027E 00	0.5192274E-01	0.9161390E 00	0.2663140E 00
33	0.1584372E 02	0.2405618E 02	0.6979618E 02	0.8835950E 02	0.1503444E 03	0.1795625E 03
34	0.1498042E 02	0.2633257E 02	0.6755036E 02	0.9387050E 02	0.1464312E 03	0.1878359E 03
35	[illegible]	-0.2881803E 01	0.4623419E 02	0.1276989E 02	0.9702817E 02	0.4086284E 02
36	-0.1080057E 01	0.3425810E 01	-0.2760761E 01	0.7254302E 01	-0.4771966E 01	0.1096441E 02
37	0.7082390E 01	-0.3029645E 01	0.2473524E 02	0.2503995E 01	0.4933448E 02	0.1351863E 02
38	[illegible]	-0.1713412E 01	0.2363105E 02	0.5261542E 01	0.4511436E 02	0.1766568E 02
39	0.1240734E 02	0.5155871E 00	0.4598653E 02	0.1989107E 02	0.9193724E 02	0.5152567E 02
40	-0.3621535E 01	0.9310263E 01	-0.7817767E [illegible]	0.1959855E 02	-0.1359972E 02	0.2946759E 02
41	[illegible]	-0.3413591E 01	0.1733661E 02	-0.1336911E 01	0.3197693E 02	0.2703396E 01
42	0.4837230E 01	-0.2107423E 01	0.1623224E 02	0.1120390E 01	0.3005637E 02	0.6854759E 01
43	0.1126603E 02	0.3912922E 01	0.4300456E 02	0.2701199E 02	0.8694569E 02	0.6220793E 02
44	[illegible]	0.1519462E 02	-0.1297531E 02	0.3192235E 02	-0.2260856E 02	0.4796979E 02
45	0.3646869E 01	-0.3328000E 01	0.9936795E 01	-0.5779425E 01	0.1591692E 02	-0.8164955E 01
46	0.3022203E 01	-0.2496918E 01	0.8832257E 01	-0.3022399E 01	0.1399603E 02	-0.3959116E 01
47	0.5886788E 01	0.8159556E 01	0.3941447E 02	0.3591302E 02	0.8068099E 02	0.7556049E 02
48	-0.7340284E 01	0.2255000E 02	-0.1919745E 02	0.4733939E 02	-0.3343253E 02	0.7109711E 02
49	0.1177285E 01	-0.4725488E 01	0.6861412E 00	-0.1095969E 02	-0.4159920E 01	-0.2162547E 02
50	0.7507823E 00	-0.2984530E 01	-0.4184704E 00	-0.2202928E 01	-0.6080976E 01	-0.1748310E 02
51	0.8847836E 01	0.1130206E 02	0.3671338E 02	0.4249976E 02	0.7577106E 02	0.8544133E 02
52	-0.9186317E 01	0.2795293E 02	-0.2387587E 02	0.5474794E 02	-0.4159036E 02	0.9821123E 02
53	-0.5035468E 00	-0.4656504E 01	-0.6159415E 01	-0.1479379E 02	-0.1901700E 02	-0.3163188E 02
54	-0.9300447E 00	-0.3345533E 01	-0.7264012E 01	-0.1203708E 02	-0.2093803E 02	-0.2748659E 02
55	-0.1349486E [illegible]	0.2407971E 01	-0.4978810E 02	-0.1275709E 02	-0.9582200E 02	-0.4077715E 02
56	[illegible]	-0.1340307E 01	0.3134839E 01	-0.7212838E 01	0.5530987E 01	-0.1075216E 02
57	0.7139[illegible]	-0.4063625E 01	0.2493651E 02	0.2526195E 01	0.4842661E 02	0.1363256E 02
58	0.4702[illegible]	-0.1737910E 01	0.2379483E 02	0.5279637E 01	0.4563336E 02	0.1775418E 02
59	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

DEAL17 Sample Problem, Input Display (Continued)

MASS MATRIX

	COLUMN 1	COLUMN 2	COLUMN 3	COLUMN 4	COLUMN 5	COLUMN 6
1	0.16666598E 04	-0.31170346E 03	0.0	0.0	0.0	0.0
2	-0.31170346E 03	0.88874806E 03	0.0	0.0	0.0	0.0
3	0.0	0.0	0.27928798E 04	-0.74911179E 03	0.0	0.0
4	0.0	0.0	-0.74911179E 03	0.20753357E 04	0.0	0.0
5	0.0	0.0	0.0	0.0	0.13474454E 04	-0.51374918E 02
6	0.0	0.0	0.0	0.0	-0.51374918E 02	0.48560458E 03
7	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0	0.0	0.0
32	0.0	0.0	0.0	0.0	0.0	0.0
33	0.0	0.0	0.0	0.0	0.0	0.0
34	0.0	0.0	0.0	0.0	0.0	0.0
35	0.0	0.0	0.0	0.0	0.0	0.0
36	0.0	0.0	0.0	0.0	0.0	0.0
37	0.0	0.0	0.0	0.0	0.0	0.0
38	0.0	0.0	0.0	0.0	0.0	0.0
39	0.0	0.0	0.0	0.0	0.0	0.0
40	0.0	0.0	0.0	0.0	0.0	0.0
41	0.0	0.0	0.0	0.0	0.0	0.0
42	0.0	0.0	0.0	0.0	0.0	0.0
43	0.0	0.0	0.0	0.0	0.0	0.0
44	0.0	0.0	0.0	0.0	0.0	0.0
45	0.0	0.0	0.0	0.0	0.0	0.0
46	0.0	0.0	0.0	0.0	0.0	0.0
47	0.0	0.0	0.0	0.0	0.0	0.0
48	0.0	0.0	0.0	0.0	0.0	0.0
49	0.0	0.0	0.0	0.0	0.0	0.0
50	0.0	0.0	0.0	0.0	0.0	0.0
51	0.0	0.0	0.0	0.0	0.0	0.0
52	0.0	0.0	0.0	0.0	0.0	0.0
53	0.0	0.0	0.0	0.0	0.0	0.0
54	0.0	0.0	0.0	0.0	0.0	0.0
55	0.0	0.0	0.0	0.0	0.0	0.0
56	0.0	0.0	0.0	0.0	0.0	0.0
57	0.0	0.0	0.0	0.0	0.0	0.0
58	0.0	0.0	0.0	0.0	0.0	0.0
60	0.0	0.0	0.0	0.0	0.0	0.0

BELL HELICOPTER COMPANY

DFAL17 Sample Problem, Input Display (Concluded)

OUTPUT DATA

DENSITY= 0.23775999E-02 REDUCED VELOCITY= 0.0

MODE	EIGENVALUE	
1	0.12305840D-01	0.0
2	0.65413330D-02	0.0
3	0.37693520D-02	0.0
4	0.17091000D-02	0.0
5	0.15267060D-02	0.0
6	0.14576990D-02	0.0
7	0.25716060D-03	0.0
8	0.21003930D-03	0.0
9	0.17888250D-03	0.0
10	0.14734150D-03	0.0
11	0.13955460D-03	0.0
12	0.11423760D-03	0.0
13	0.90607680D-04	0.0
14	0.84456430D-04	0.0
15	0.40531410D-04	0.0

DFAL17 Sample Problem, Condensed Output

MODE	OMEGA (CPS)	DAMPING	VELOCITY (KNOTS)
1	0.14347105E 01	0.0	0.0
2	0.19102869E 01	0.0	0.0
3	0.25973090E 01	0.0	0.0
4	0.37690525E 01	0.0	0.0
5	0.40598922E 01	0.0	0.0
6	0.41829138E 01	0.0	0.0
7	0.92326397E 01	0.0	0.0
8	0.10981708E 02	0.0	0.0
9	0.11899706E 02	0.0	0.0
10	0.13111655E 02	0.0	0.0
11	0.13472487E 02	0.0	0.0
12	0.14890707E 02	0.0	0.0
13	0.16720047E 02	0.0	0.0
14	0.17097183E 02	0.0	0.0
15	0.24876633E 02	0.0	0.0

DFAL17 Sample Problem, Condensed Output (Continued)

EIGENVECTORS

	1 VECTOR		2 VECTOR		3 VECTOR	
1	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
2	[illegible]	0.0	0.5043449D-02	0.0	-0.3357114D-01	0.0
3	0.1660189D-01	0.0	0.4488670D-01	0.0	[illegible]	0.0
4	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
5	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
6	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
7	[illegible]	0.0	0.1478747	0.0	[illegible]	0.0
8	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
9	[illegible]	0.0	0.2272544	0.0	[illegible]	0.0
10	[illegible]	0.0	[illegible]	0.0	-0.2307414	0.0
11	0.1141356	0.0	[illegible]	0.0	[illegible]	0.0
12	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
13	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
14	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
15	-0.4591216D-04	0.0	0.1579914D-02	0.0	[illegible]	0.0
16	[illegible]	0.0	0.3234604D-02	0.0	[illegible]	0.0
17	0.2028243	0.0	0.3423633	0.0	0.3625417	0.0
18	0.2032656	0.0	0.3292593	0.0	[illegible]	0.0
19	[illegible]	0.0	0.1128861D-02	0.0	[illegible]	0.0
20	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
21	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
22	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
23	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
24	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
25	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
26	0.5455933	0.0	0.5053369D-01	0.0	[illegible]	0.0
27	-0.4642550D-03	0.0	[illegible]	0.0	[illegible]	0.0
28	[illegible]	0.0	0.5815929D-02	0.0	[illegible]	0.0
29	[illegible]	0.0	-0.1216024	0.0	[illegible]	0.0
30	0.7956472	0.0	-0.1376548	0.0	-0.1147979	0.0
31	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
32	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
33	0.9888313	0.0	[illegible]	0.0	-0.1477540	0.0
34	1.000000	0.0	-0.2824677	0.0	[illegible]	0.0
35	0.5875585D-01	0.0	[illegible]	0.0	[illegible]	0.0
36	0.2290142D-03	0.0	-0.1459369D-01	0.0	[illegible]	0.0
37	[illegible]	0.0	0.2717351	0.0	[illegible]	0.0
38	-0.1154971D-01	0.0	[illegible]	0.0	[illegible]	0.0
39	0.4923909D-01	0.0	[illegible]	0.0	[illegible]	0.0
40	[illegible]	0.0	-0.4392532D-01	0.0	-0.3224435	0.0
41	-0.1887412	0.0	[illegible]	0.0	0.7534411D-01	0.0
42	-0.1904254	0.0	0.3913967	0.0	[illegible]	0.0
43	0.5972984D-01	0.0	0.3222852	0.0	[illegible]	0.0
44	0.1839740D-02	0.0	[illegible]	0.0	[illegible]	0.0
45	-0.3583662	0.0	[illegible]	0.0	[illegible]	0.0
46	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
47	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
48	0.2566744D-02	0.0	[illegible]	0.0	[illegible]	0.0
49	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
50	-0.5956740	0.0	[illegible]	0.0	[illegible]	0.0
51	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
52	[illegible]	0.0	-0.1363035	0.0	[illegible]	0.0
53	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
54	-0.7852338	0.0	1.000000	0.0	[illegible]	0.0
55	[illegible]	0.0	-0.3544719	0.0	[illegible]	0.0
56	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
57	-0.3169456D-01	0.0	[illegible]	0.0	[illegible]	0.0
58	[illegible]	0.0	0.2679929	0.0	0.1219092	0.0
59	[illegible]	0.0	[illegible]	0.0	[illegible]	0.0
60	-0.1204358D-02	0.0	0.4831903D-01	0.0	[illegible]	0.0
61	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

DFAL17 Sample Problem, Condensed Output (Concluded)

```
DFAL18      SAMPLE  PROBLEM

DATA GUST CASE   SINE-SQUARED    10 FT   30 FOLD   175 KNOTS          CARD 01
 -2    2    8    1    1                                                CARD 02
.002378        175.           5.7          10.          4.88     97.075   CARD 03
.005           1.33           5.                  1.5-6-7.5        30.      CARD 04
960000000000.2200000000000.55.             110.                           CARD 05
4.15           2.73           2.53         2.6          2.71     0.       CARD 06
0.             0              3.93         4.42         3.33     2.5      CARD 07
0.             1.965          2.21         1.665        1.25     0.       CARD 08
1.965          2.21           1.665        1.25                           CARD 09
5.75           5.926          5.16         4.88         4.59     1.       CARD 10
1.             1.52           1.9          2.           2.       1.75     CARD 11
1.52           1.9            2.           2.           1.75     1.52     CARD 12
1.9            2.             2.           1.75                           CARD 13
.77            1.72           .195         1.88         .011     2.54  }
.002           3.77           .341         4.7          .324     4.8   }  CARD 14
.101           9.45           .00997       16.8                        }
```

DFAL18 Sample Problem, Input Data

.99452	4.88	97.075	603.	11.2		CARD 15
4.15	2.73	2.43	2.6	2.71	0.	CARD 16
0.						
5.75	5.426	5.16	4.88	4.59	1.	CARD 17
1.						
5.6	5.6	5.6	5.6	5.6	1.	CARD 18
1.						
.86603	4.88	97.075	603.	11.2		CARD 19
0.	3.93	4.42	3.33	2.5		CARD 20
1.52	1.9	2.	2.	1.75		CARD 21
1.667	1.667	1.667	1.667	1.667		CARD 22
.86603	4.88	97.075	603.	11.2		CARD 23
0.	3.93	4.42	3.33	2.5		CARD 24
1.52	1.9	2.	2.	1.75		CARD 25

DATA SHEET

DFAL18 Sample Problem, Input Data (Continued)

1.667	1.667	1.667	1.667	1.667		CARD 26
.86603	4.88	97.075	603.	11.2		CARD 27
0.	3.93	4.42	3.33	2.5		CARD 28
1.52	1.9	2.	2.	1.75		CARD 29
1.667	1.667	1.667	1.667	1.667		CARD 30
13.4	13.4	13.4	13.4	13.4	122.0	CARD 31
10.	.1	.1	.1	.1		CARD 32
53.6	53.6	53.6	53.6	53.6	10.	CARD 33
10.	20.	20.	20.	20.	20.	CARD 34
1930.	3370.	1450.	2950.	1450.	4187.	CARD 35
4187.	170.	213.2	105.7	108.5	46.	CARD 36
12800.	43700.	9650.	38050.	9650.	-390000.	CARD 37
10.	.1	.1	.1	.1	.1	CARD 38
3300000.	7540000.	1660000.	6250000.	120000.	44800000.	CARD 39
44800000.	100.	42200.	8700.	12500.	9650.	CARD 40

DFAL18 Sample Problem, Input Data (Continued)

```
FLEX MATRIX (74X74)        CARD 41
EIGENVALUES -74            CARD 42
```

DATA SHEET

DFAL18 Sample Problem, Input Data (Concluded)

GUST RESPONSE ANALYSIS FOR FOLDING PROPROTOR

D270 GUST CASE   SINE-SQUARED   10 FT   30 FOLD   175 KNOTS

CONTROL DATA

GUST NUMBER	-2
OPTION NUMBER	2
NO. OF FLUTTER FREQUENCIES	8
NO. OF SECTIONS	1
DENSITY,SLUG/FT**3	0.23780E-02
AIRCRAFT VELOCITY,KT	0.17500E 03
LIFT CURVE SLOPE	0.57000E 01
GUST VELOCITY	0.10000E 02
REF. SEMICHORD,FT	0.46800E 01
SPAN,FT	0.27075E 02
TIME INCREMENT,SEC	0.50000D-02
GUST FREQUENCY,CPS	0.13300E 01
MAX. RUN TIME,SEC	0.50000E 01
FLEXIBILITY FACTOR	0.10000D-05
WING SWEEP,DEG	-0.75000E 01
BLADE SWEEP,DEG	0.30000E 02

SECTION PARAMETERS

SECTION	EI,LB-IN**2	GJ,LB-IN**2	L1,IN	L2,IN
1	0.46000E 11	0.22000E 11	0.55000E 02	0.11000E 03

	STRIP WIDTHS(GUST)	SEMICHORDS(GUST)
1	0.41500E 01	0.57500E 01
2	0.27300E 01	0.54260E 01
3	0.24300E 01	0.51600E 01
4	0.26000E 01	0.48800E 01
5	0.27100E 01	0.45900E 01
6	0.0	0.10000E 01
7	0.0	0.10000E 01
8	0.0	0.15200E 01
9	0.39300E 01	0.19000E 01
10	0.44200E 01	0.20000E 01
11	0.33300E 01	0.20000E 01
12	0.15000E 01	0.17500E 01
13	0.0	0.15200E 01
14	0.19650E 01	0.19000E 01
15	0.22100E 01	0.20000E 01
16	0.16550E 01	0.20000E 01
17	0.12500E 01	0.17500E 01
18	0.0	0.15200E 01
19	0.19650E 01	0.19000E 01
20	0.22100E 01	0.20000E 01
21	0.16650E 01	0.20000E 01
22	0.12500E 01	0.17500E 01

DFAL18 Sample Problem, Input Display

INPUT FREQUENCY-DAMPING VALUES

MODE	DAMPING	FREQUENCY
1	0.77000D 00	0.17200D 01
2	0.14530D 00	0.18800D 01
3	0.11000D-01	0.25400D 01
4	0.20000D-02	0.37700D 01
5	0.34100D 00	0.47000D 01
6	0.32400D 00	0.48000D 01
7	0.10100D 00	0.94500D 01
8	0.99700D-02	0.10800D 02

AERODYNAMIC INPUT DATA

SURFACE	COSINE LAMBDA	BR	S	C*S	C BAR
1	0.99452E 00	0.48800E 01	0.57075E 02	0.50300E 03	0.11200E 02
2	0.86603E 00	0.48800E 01	0.57075E 02	0.50300E 03	0.11200E 02
3	0.86603E 00	0.48800E 01	0.57075E 02	0.50300E 03	0.11200E 02
4	0.86603E 00	0.48800E 01	0.57075E 02	0.50300E 03	0.11200E 02

	STRIP	1	2	3	4	5	6	7
SURFACE 1	CV	0.41500E 01	0.27300E 01	0.24300E 01	0.26300E 01	0.27100E 01	0.0	0.0
	B	0.57500E 01	0.54260E 01	0.51600E 01	0.48900E 01	0.45900E 01	0.10000E 01	0.10000E 01
	D	0.55000E 01	0.56000E 01	0.56000E 01	0.56000E 01	0.56000E 01	0.10000E 01	0.10000E 01
SURFACE 2	CV	0.0	0.39300E 01	0.44200E 01	0.33300E 01	0.25000E 01		
	B	0.15200E 01	0.19000E 01	0.20000E 01	0.20000E 01	0.17500E 01		
	D	0.16670E 01	0.16670E 01	0.16670E 01	0.16670E 01	0.16670E 01		
SURFACE 3	CV	0.0	0.39300E 01	0.44200E 01	0.33300E 01	0.25000E 01		
	B	0.15200E 01	0.19000E 01	0.20000E 01	0.20000E 01	0.17500E 01		
	D	0.16670E 01	0.16670E 01	0.16670E 01	0.16670E 01	0.16670E 01		
SURFACE 4	CV	0.0	0.39300E 01	0.44200E 01	0.33300E 01	0.25000E 01		
	B	0.15200E 01	0.19000E 01	0.20000E 01	0.20000E 01	0.17500E 01		
	D	0.16670E 01	0.16670E 01	0.16670E 01	0.16670E 01	0.16670E 01		

DFAL18 Sample Problem, Input Display (Continued)

FLEXIBILITY MATRIX, IN/LB

0.1301D 01	-0.7637D 00	0.3314D 01	0.1192D 01	0.5542D 01	0.2353D 01	0.7412D 01	0.4561D 01	0.9273D 01	0.6441D 01	0.1569D 02
0.1002D 02	0.3495D-01	0.6110D-01	0.3312D-01	0.5743D-01	0.1530D 02	0.1543D 02	0.2383D-01	0.5334D-01	0.1619D 02	0.1533D 02
0.1466D-01	0.9936D-01	0.1504D 02	0.1522D 02	0.3129D-02	0.1133D 00	0.1594D 02	0.1508D 02	-0.6487D-02	0.1341D 00	0.1564D 02
0.1456D 02	0.1353D 02	-0.1039D 01	0.7062D 01	0.6056D 01	0.1241D 02	-0.3021D 01	0.5246D 01	0.4534D 01	0.1129D 02	-0.4963D 01
0.3449D 01	0.3022D 01	0.4365D 01	-0.7393D 01	0.1177D 01	0.7537D 00	0.4445D 01	-0.9146D 01	-0.5035D 00	-0.9301D 00	-0.1349D 02
0.1157D 01	0.7143D 01	0.6703D 01	-0.1233D 02	0.3112D 01	0.5337D 01	0.4470D 01	-0.1127D 02	0.5034D 01	0.3474D 01	0.3037D 01
-0.9882D 01	0.7444D 01	0.1133D 01	0.7455D 00	-0.6554D 01	0.3226D 01	-0.5129D 00	-0.9501D 00			
-0.7637D 00	0.7870D 01	0.5519D 00	0.9205D 01	0.1532D 01	0.1331D 02	0.2645D 01	0.1149D 02	0.3845D 01	0.1258D 02	-0.3581D 01
0.1357D 02	-0.2233D-01	-0.2550D-01	-0.1343D-01	-0.2413D-01	-0.1354D 01	0.5592D 00	-0.5475D-02	-0.3450D-01	0.4309D 01	0.6935D 01
-0.6122D-02	-0.4140D-01	0.1055D 02	0.1320D 02	-0.1333D-02	-0.4435D-01	0.1441D 02	0.2104D 02	0.2265D-02	-0.5601D-01	0.2421D 02
0.2553D 02	-0.2892D 01	0.3425D 01	-0.3030D 01	-0.1719D 01	0.5153D 00	0.9315D 01	-0.3417D 01	-0.2107D 01	0.3913D 01	0.1519D 02
-0.3838D 01	-0.2497D 01	0.6159D 01	0.2255D 02	-0.4245D 01	-0.2934D 01	0.1113D 02	0.2740D 02	-0.4455D 01	-0.3345D 01	0.2869D 01
-0.3470D 01	-0.3054D 01	-0.1733D 01	-0.5253D 00	-0.9343D 01	-0.3435D 01	-0.2123D 01	-0.3919D 01	-0.1523D 02	-0.3519D 01	-0.2503D 01
-0.8151D 01	-0.2257D 02	-0.4245D 01	-0.2942D 01	-0.1130D 02	-0.2431D 02	-0.4552D 01	-0.3537D 01			
0.3514D 01	0.5519D 00	0.1203D 02	0.5414D 01	0.1423D 02	0.1243D 02	0.2561D 02	0.1957D 02	0.3473D 02	0.2744D 02	0.5663D 02
0.4158D 02	0.1327D 00	0.2142D 00	0.1161D 00	0.2353D 00	0.5474D 02	0.5754D 02	0.6374D-01	0.2923D 00	0.6218D 02	0.5994D 02
0.5140D-01	0.3465D 00	0.4456D 02	0.6234D 02	0.1397D-01	0.4133D 00	0.6755D 02	0.6533D 02	-0.1396D-01	0.4701D 00	0.6980D 02
0.6755D 02	0.4690D 02	-0.2761D 01	0.2474D 02	0.2363D 02	0.4594D 02	-0.7317D 01	0.1734D 02	0.1623D 02	0.4396D 02	-0.1287D 02
0.9937D 01	0.8532D 01	0.3941D 02	-0.1923D 02	0.6460D 00	-0.4155D 00	0.3571D 02	-0.2387D 02	-0.6153D 01	-0.7264D 01	-0.4979D 02
0.3135D 01	0.2494D 02	0.2373D 02	-0.4590D 02	0.6135D 01	0.1744D 02	0.1534D 02	-0.4301D 02	0.1314D 02	0.1003D 02	0.8864D 01
-0.3748D 02	0.1935D 02	0.7352D 00	-0.4369D 00	-0.3473D 02	0.2431D 02	-0.5192D 01	-0.7334D 01			
0.1192D 01	0.5206D 01	0.5414D 01	0.2433D 02	0.1115D 02	0.2451D 02	0.1590D 02	0.3532D 02	0.2417D 02	0.4156D 02	0.1419D 02
0.5036D 02	0.2133D-01	0.2371D-01	0.1283D-01	0.2140D-01	0.1955D 02	0.2556D 02	0.4235D-02	0.3233D-01	0.3603D 02	0.4154D 02
0.5674D-02	0.3547D-01	0.5251D 02	0.5802D 02	0.1223D-02	0.4513D-01	0.7311D 02	0.7562D 02	-0.2371D-02	0.5148D-01	0.8835D 02
0.9357D 02	0.1277D 02	0.7254D 01	0.2504D 01	0.5261D 01	0.1344D 02	0.1959D 02	-0.1637D 01	0.1123D 01	0.2701D 02	0.3192D 02
-0.5730D 01	-0.3022D 01	0.3591D 02	0.4734D 02	-0.1395D 02	-0.3234D 01	0.4290D 02	0.5675D 02	-0.1479D 02	-0.1234D 02	-0.1276D 02
-0.7213D 01	0.2525D 01	0.5274D 01	-0.1445D 02	-0.1455D 02	-0.1521D 01	0.1132D 01	-0.2781D 02	-0.3189D 02	-0.5770D 01	-0.3017D 01
-0.3594D 02	-0.4732D 02	-0.1095D 02	-0.4205D 01	-0.4250D 02	-0.5971D 02	-0.1490D 02	-0.1204D 02			
0.5542D 01	0.1532D 01	0.1923D 02	0.1115D 02	0.3432D 02	0.2413D 02	0.5079D 02	[illegible]	0.6644D 02	0.5565D 02	0.1123D 03
0.3450D 02	0.3755D 00	0.4175D 00	0.2252D 00	[illegible]	0.1147D 03	0.1157D 03	0.1532D 00	0.5595D 00	0.1270D 03	0.1231D 03
0.1031D 00	0.6757D 00	0.1344D 03	0.1334D 03	0.2134D-01	0.8151D 00	0.1435D 03	0.1446D 03	-0.3689D-01	0.9140D 00	0.1503D 03
0.1464D 03	0.9733D 02	-0.4772D 01	0.4304D 02	0.4110D 02	0.9144D 02	-0.1359D 02	0.4194D 02	0.3005D 02	0.6685D 02	-0.2241D 02
0.1592D 02	0.1403D 02	0.6543D 02	-0.3343D 02	-0.4153D 01	-0.6341D 01	0.7577D 02	-0.4159D 02	-0.1902D 02	-0.2094D 02	-0.9680D 02
0.5501D 01	0.4943D 02	0.4543D 02	-0.9177D 02	0.1421D 02	[illegible]	0.3327D 02	-0.8675D 02	0.2292D 02	0.1509D 02	0.1410D 02
-0.8345D 02	0.3340D 02	-0.4123D 01	-0.6117D 01	-0.7581D 02	0.4145D 02	-0.1774D 02	-0.2107D 02			
0.2460D 01	0.1331D 02	0.1243D 02	0.2461D 02	0.2313D 02	0.5111D 02	0.3545D 02	0.6424D 02	0.5103D 02	0.7869D 02	0.4630D 02
0.1306D 03	0.1033D 00	0.1215D 03	0.6570D-01	0.1440D 00	0.5735D 02	0.5542D 02	0.4740D-01	0.1555D 00	0.9639D 02	0.9465D 02
0.2911D-01	0.1973D 00	0.1157D 03	0.1240D 03	0.6245D-02	0.2354D 00	0.1524D 03	0.1607D 03	-0.1086D-01	0.2562D 00	0.1795D 03
0.1575D 03	0.4084D 02	0.1095D 02	0.1352D 02	0.1765D 02	0.5152D 02	0.2947D 02	0.2704D 01	0.5554D 01	0.6221D 02	0.4797D 02
-0.6105D 01	-0.3994D 01	0.7555D 02	0.7110D 02	-0.2154D 02	-0.1744D 02	0.5544D 02	0.4521D 02	-0.3163D 02	-0.2749D 02	-0.4073D 02
-0.1075D 02	0.1363D 02	0.1774D 02	-0.5146D 02	[illegible]	0.2743D 01	0.6915D 01	-0.5214D 02	-0.4742D 02	-0.8055D 01	-0.3933D 01
-0.7555D 02	-0.7099D 02	-0.2151D 02	-0.1744D 02	-0.3145D 02	-0.3813D 02	-0.3165D 02	-0.2753D 02			
0.7312D 01	0.2595D 01	0.2561D 02	0.1690D 02	0.5579D 02	0.3345D 02	0.6037D 02	0.6112D 02	0.1127D 03	0.9274D 02	0.1897D 03
0.1495D 03	0.6225D 00	0.6921D 00	0.3744D 00	0.7434D 00	0.2341D 03	0.1953D 03	0.2704D 00	0.9442D 00	0.2205D 03	0.2144D 03
0.1655D 00	0.1125D 01	0.2369D 03	0.2504D 03	0.3550D-01	0.1551D 01	0.2574D 03	0.2514D 03	-0.6103D-01	0.1516D 01	0.2725D 03
0.2564D 03	0.1540D 03	-0.7427D 01	0.7928D 02	0.7623D 02	0.1555D 03	-0.2122D 02	0.4423D 02	0.4623D 02	0.1480D 03	-0.3502D 02
0.1927D 02	0.1527D 02	0.1341D 03	-0.5227D 02	-0.1424D 02	-0.2124D 02	0.1337D 03	-0.6503D 02	-0.4603D 02	-0.4900D 02	-0.1636D 03
0.6637D 01	0.7993D 02	0.7551D 02	-0.1557D 03	0.2225D 02	0.4775D 02	0.4554D 02	-0.1474D 03	0.3547D 02	0.1956D 02	0.1644D 02
-0.1340D 03	0.5248D 02	-0.1514D 02	-0.2140D 02	-0.1363D 03	0.6544D 02	-0.4610D 02	-0.4923D 02			
0.4561D 01	0.1143D 02	0.1957D 02	0.3533D 02	0.3462D 02	0.6429D 02	0.6112D 02	0.9923D 02	0.9907D 02	0.1266D 03	0.9655D 02
0.1702D 03	0.2478D 00	0.2754D 00	0.1490D 00	0.3337D 00	0.1155D 03	0.1254D 03	0.1075D 00	0.3755D 00	0.1624D 03	0.1741D 03
0.6600D-01	0.4474D 00	0.2102D 03	0.2214D 03	0.1416D-01	0.5372D 00	0.2644D 03	0.2405D 03	-0.2423D-01	0.5036D 00	0.3132D 03
0.3244D 03	0.6465D 02	0.1496D 02	0.3073D 02	0.3537D 02	0.9913D 02	0.4092D 02	0.6533D 01	0.1415D 02	0.1135D 03	0.6508D 02

DFAL18 Sample Problem, Input Display (Continued)

MASS INPUT DATA

SEGMENT	X	Y	M	SX	IX
1	0.13400D 02	0.53600D 02	0.19300D 04	0.12900D 05	0.33000D 07
2	0.13400D 02	0.53600D 02	0.33700D 04	0.43700D 05	0.75400D 07
3	0.13400D 02	0.53600D 02	0.14500D 04	0.96500D 04	0.16600D 07
4	0.13400D 02	0.53600D 02	0.29500D 04	0.38050D 05	0.62500D 07
5	0.13400D 02	0.53600D 02	0.14500D 04	0.96500D 04	0.12000D 06
6	0.12200D 03	0.10000D 02	0.41870D 04	-0.39000D 06	0.44800D 08
7	0.10000D 02	0.10000D 02	0.41870D 04	0.10000D 02	0.44800D 08
8	0.10000D 00	0.20000D 02	0.17000D 03	0.10000D 00	0.10000D 03
9	0.10000D 00	0.20000D 02	0.21320D 03	0.10000D 00	0.42200D 05
10	0.10000D 00	0.20000D 02	0.10570D 03	0.10000D 00	0.87000D 04
11	0.10000D 00	0.20000D 02	0.10850D 03	0.10000D 00	0.12500D 05
12	0.0	0.20000D 02	0.46000D 02	0.10000D 00	0.96500D 04
13	0.10000D 00	0.20000D 02	0.17000D 03	0.10000D 00	0.10000D 03
14	0.10000D 00	0.20000D 02	0.21320D 03	0.10000D 00	0.42200D 05
15	0.10000D 00	0.20000D 02	0.10570D 03	0.10000D 00	0.87000D 04
16	0.10000D 00	0.20000D 02	0.10850D 03	0.10000D 00	0.12500D 05
17	0.0	0.20000D 02	0.46000D 02	0.10000D 00	0.96500D 04
18	0.10000D 00	0.20000D 02	0.17000D 03	0.10000D 00	0.10000D 03
19	0.10000D 00	0.20000D 02	0.21320D 03	0.10000D 00	0.42200D 05
20	0.10000D 00	0.20000D 02	0.10570D 03	0.10000D 00	0.87000D 04
21	0.10000D 00	0.20000D 02	0.10850D 03	0.10000D 00	0.12500D 05
22	0.0	0.20000D 02	0.46000D 02	0.10000D 00	0.96500D 04

FLUTTER SYSTEM ROOTS

1	0.85522D-02	0.65929D-02	2	0.71654D-02	0.10392D-02	3	0.39262D-02	0.43199D-04
4	0.17822D-02	0.35644D-05	5	0.11467D-02	0.39102D-03	6	0.10994D-02	0.35521D-03
7	0.23365D-03	0.28648D-04	8	0.21717D-03	0.21651D-05	9	0.17848D-03	0.0
10	0.14734D-03	0.0	11	0.13955D-03	0.0	12	0.11424D-03	0.0
13	0.90508D-04	0.0	14	0.86654D-04	0.0	15	0.40951D-04	0.0
16	0.40216D-04	0.0	17	0.39309D-04	0.0	18	0.22437D-04	0.0
19	0.22141D-04	0.0	20	0.19522D-04	0.0	21	0.10599D-04	0.0
22	0.99944D-05	0.0	23	0.64572D-05	0.0	24	0.57163D-05	0.0
25	0.56579D-05	0.0	26	0.53422D-05	0.0	27	0.51499D-05	0.0
28	0.50793D-05	0.0	29	0.36559D-05	0.0	30	0.21410D-05	0.0
31	0.12952D-05	0.0	32	0.90403D-06	0.0	33	0.90907D-06	0.0
34	0.90539D-06	0.0	35	0.88922D-06	0.0	36	0.88722D-06	0.0
37	0.87694D-06	0.0	38	0.63654D-06	0.0	39	0.42446D-06	0.0
40	0.42442D-06	0.0	41	0.39947D-06	0.0	42	0.27438D-06	0.0
43	0.27423D-06	0.0	44	0.27349D-06	0.0	45	0.23961D-06	0.0
46	0.21902D-06	0.0	47	0.21877D-06	0.0	48	0.18604D-06	0.0
49	0.12092D-06	0.0	50	0.12092D-06	0.0	51	0.11952D-06	0.0
52	0.22691D-08	0.0	53	0.21413D-08	0.0	54	0.14267D-08	0.0
55	0.15998D-08	0.0	56	0.13179D-08	0.0	57	0.11406D-09	0.0
58	0.10206D-09	0.0	59	0.37160D-10	0.0	60	0.35082D-10	0.0
61	0.31634D-10	0.0	62	0.79250D-13	0.0	63	0.13755D-14	0.0
64	0.13763D-14	0.0	65	0.13502D-14	0.0	66	0.42169D-15	0.0
67	0.42147D-15	0.0	68	0.41792D-15	0.0	69	0.21096D-15	0.0
70	0.21095D-15	0.0	71	0.21085D-15	0.0	72	0.12245D-15	0.0
73	0.12229D-15	0.0	74	0.12212D-15	0.0			

DFAL18 Sample Problem, Input Display (Concluded)

0270 GUST CASE SINE-SQUARED 10 FT 30 FOLD 175 KNOTS

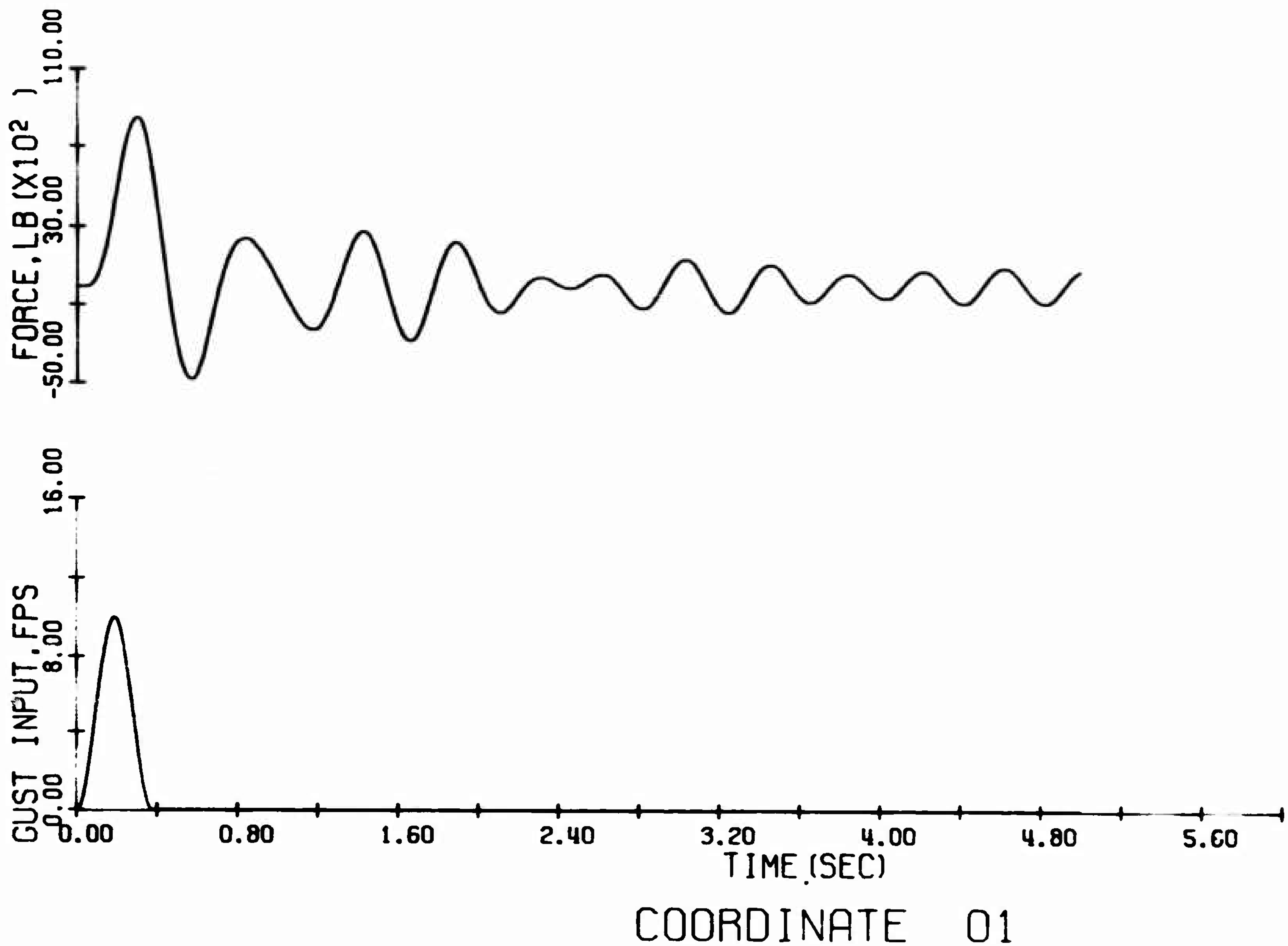
SYSTEM RESPONSES

COORDINATE 01

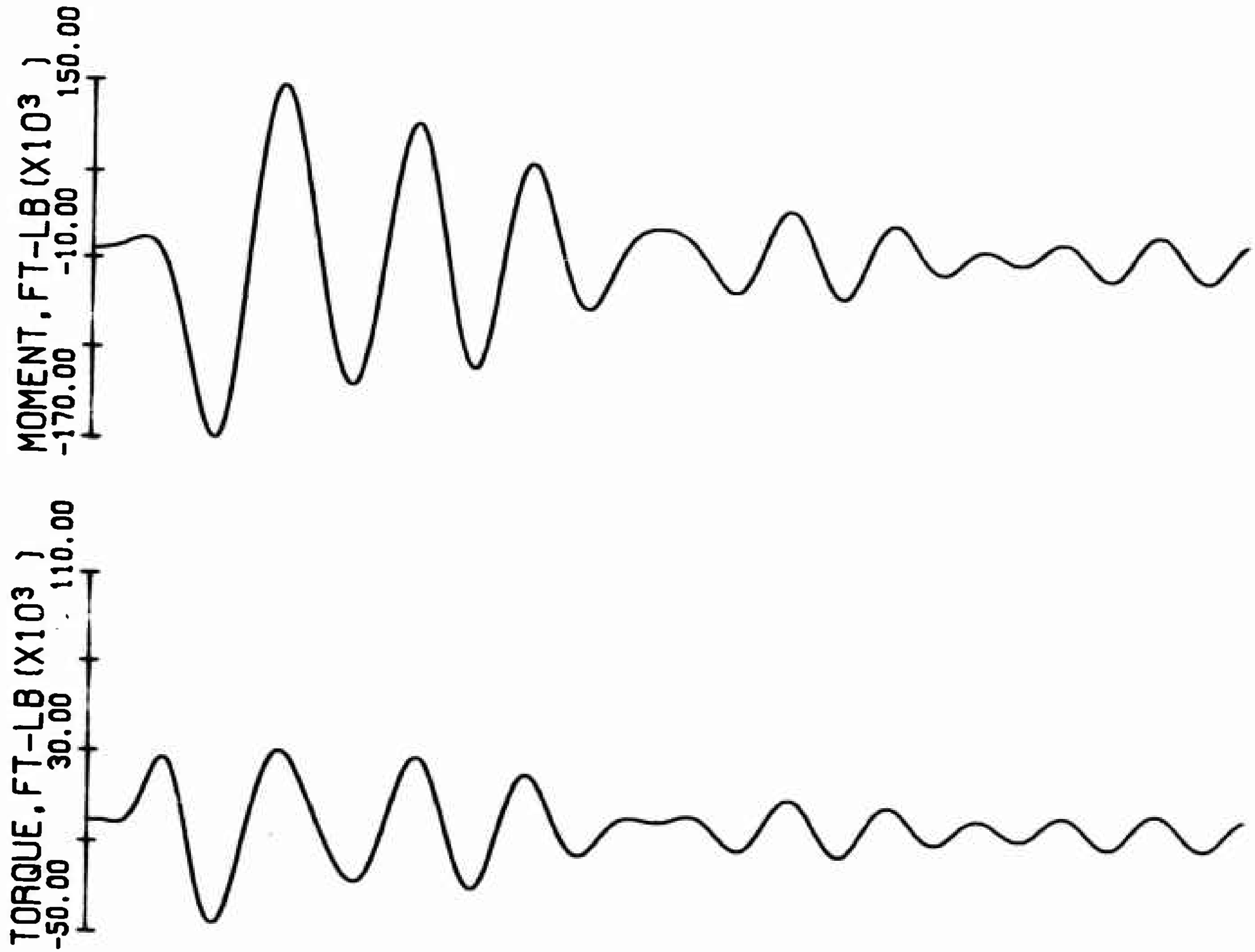
FORCE,LB

0.0	-0.15591E-02	-0.11753E-01	-0.36352E-01	-0.69643E-01	-0.33360E-01	0.24244E 00	0.11013E 01	0.30842E 01	0.69713E 01
0.13740E 02	0.24532E 02	0.41459E 02	0.55340E 02	0.59064E 02	0.14120E 03	0.19616E 03	0.26412E 03	0.34594E 03	0.44215E 03
0.55254E 03	0.57745E 03	0.81686E 03	0.96442E 03	0.11133E 04	0.13091E 04	0.14956E 04	0.16922E 04	0.18983E 04	0.21139E 04
0.23346E 04	0.25731E 04	0.28170E 04	0.30705E 04	0.33337E 04	0.35053E 04	0.38874E 04	0.41760E 04	0.44707E 04	0.47693E 04
0.50595E 04	0.53639E 04	0.56644E 04	0.59534E 04	0.62332E 04	0.65014E 04	0.67559E 04	0.69952E 04	0.72141E 04	0.74239E 04
0.76123E 04	0.77935E 04	0.79371E 04	0.80739E 04	0.81448E 04	0.82455E 04	0.83820E 04	0.84493E 04	0.84975E 04	0.85251E 04
0.85305E 04	0.85125E 04	0.84683E 04	0.83596E 04	0.83325E 04	0.81775E 04	0.80244E 04	0.78449E 04	0.76390E 04	0.74089E 04
0.71505E 04	0.68345E 04	0.65551E 04	0.62900E 04	0.59735E 04	0.56457E 04	0.53049E 04	0.49544E 04	0.46134E 04	0.42565E 04
0.38443E 04	0.35275E 04	0.31564E 04	0.27515E 04	0.24334E 04	0.20237E 04	0.16410E 04	0.12597E 04	0.87723E 03	0.49919E 03
0.12311E 03	-0.24651E 03	-0.60352E 03	-0.96173E 03	-0.13044E 04	-0.15351E 04	-0.19526E 04	-0.22554E 04	-0.25437E 04	-0.28156E 04
-0.30711E 04	-0.33304E 04	-0.35314E 04	-0.37359E 04	-0.39231E 04	-0.40933E 04	-0.42455E 04	-0.43805E 04	-0.44979E 04	-0.45975E 04
-0.46789E 04	-0.47417E 04	-0.47856E 04	-0.48107E 04	-0.48162E 04	-0.48020E 04	-0.47630E 04	-0.47143E 04	-0.46412E 04	-0.45491E 04
-0.44357E 04	-0.43110E 04	-0.41670E 04	-0.40082E 04	-0.38359E 04	-0.35514E 04	-0.34574E 04	-0.32544E 04	-0.30444E 04	-0.28288E 04
-0.25951E 04	-0.23435E 04	-0.21514E 04	-0.19354E 04	-0.17105E 04	-0.14651E 04	-0.12635E 04	-0.10373E 04	-0.81537E 03	-0.59680E 03
-0.36062E 03	-0.15770E 03	0.43044E 02	0.24377E 03	0.44114E 03	0.63153E 03	0.81431E 03	0.99430E 03	0.11524E 04	0.13071E 04
0.14563E 04	0.15636E 04	0.17352E 04	0.18150E 04	0.19150E 04	0.20059E 04	0.20651E 04	0.21573E 04	0.22201E 04	0.22752E 04
0.23232E 04	0.23544E 04	0.23942E 04	0.24275E 04	0.24501E 04	0.24662E 04	0.24750E 04	0.24791E 04	0.24755E 04	0.24650E 04
0.24475E 04	0.24232E 04	0.23922E 04	0.23544E 04	0.23115E 04	0.22624E 04	0.22035E 04	0.21520E 04	0.20910E 04	0.20272E 04
0.19610E 04	0.18924E 04	0.18231E 04	0.17517E 04	0.16784E 04	0.16044E 04	0.15241E 04	0.14495E 04	0.13693E 04	0.12954E 04
0.12009E 04	0.11125E 04	0.10224E 04	0.92955E 03	0.83432E 03	0.73773E 03	0.63956E 03	0.54035E 03	0.44094E 03	0.34112E 03
0.24143E 03	0.14211E 03	0.43307E 02	-0.56445E 02	-0.15251E 03	-0.24465E 03	-0.34425E 03	-0.44255E 03	-0.53946E 03	-0.63459E 03
-0.73331E 03	-0.82575E 03	-0.92376E 03	-0.10154E 04	-0.11035E 04	-0.12005E 04	-0.12930E 04	-0.13740E 04	-0.14646E 04	-0.15473E 04
-0.15267E 04	-0.17326E 04	-0.17745E 04	-0.14424E 04	-0.14060E 04	-0.13651E 04	-0.23135E 04	-0.20591E 04	-0.21136E 04	-0.21529E 04
-0.21455E 04	-0.22145E 04	-0.22363E 04	-0.22517E 04	-0.22533E 04	-0.22613E 04	-0.22554E 04	-0.22425E 04	-0.22214E 04	-0.21923E 04
-0.21332E 04	-0.21100E 04	-0.22059E 04	-0.14454E 04	-0.14270E 04	-0.13503E 04	-0.17651E 04	-0.16744E 04	-0.15755E 04	-0.14594E 04
-0.13505E 04	-0.12370E 04	-0.11111E 04	-0.97715E 03	-0.84157E 03	-0.69877E 03	-0.55124E 03	-0.39970E 03	-0.24462E 03	-0.86729E 02
0.73255E 02	0.23450E 03	0.39558E 03	0.55445E 03	0.71452E 03	0.87912E 03	0.10355E 04	0.11912E 04	0.13425E 04	0.14899E 04
0.16326E 04	0.17772E 04	0.19028E 04	0.20255E 04	0.21474E 04	0.22593E 04	0.23627E 04	0.24572E 04	0.25423E 04	0.26173E 04
0.26817E 04	0.27350E 04	0.27767E 04	0.28057E 04	0.28245E 04	0.28309E 04	0.28234E 04	0.28054E 04	0.27750E 04	0.27329E 04
0.26743E 04	0.26144E 04	0.25337E 04	0.24522E 04	0.23554E 04	0.22455E 04	0.21314E 04	0.20050E 04	0.18702E 04	0.17262E 04
0.15734E 04	0.14134E 04	0.12465E 04	0.10735E 04	0.89473E 03	0.71134E 03	0.52447E 03	0.33495E 03	0.14399E 03	-0.47712E 02
-0.23095E 03	-0.42636E 03	-0.61540E 03	-0.80322E 03	-0.94535E 03	-0.11562E 04	-0.13255E 04	-0.14409E 04	-0.15437E 04	-0.17993E 04
-0.19421E 04	-0.20795E 04	-0.22019E 04	-0.23176E 04	-0.24213E 04	-0.25175E 04	-0.25604E 04	-0.26713E 04	-0.27244E 04	-0.27750E 04
-0.24073E 04	-0.28252E 04	-0.29317E 04	-0.28235E 04	-0.24030E 04	-0.27652E 04	-0.27226E 04	-0.26544E 04	-0.25943E 04	-0.25130E 04
-0.24210E 04	-0.23137E 04	-0.22057E 04	-0.20355E 04	-0.14555E 04	-0.14175E 04	-0.15713E 04	-0.15191E 04	-0.13601E 04	-0.11955E 04
-0.10261E 04	-0.85294E 03	-0.67650E 03	-0.49270E 03	-0.31365E 03	-0.14065E 03	0.37272E 02	0.21317E 03	0.38610E 03	0.55521E 03
0.71970E 03	0.87315E 03	0.10320E 04	0.11765E 04	0.13179E 04	0.14449E 04	0.15732E 04	0.16932E 04	0.17941E 04	0.18936E 04
0.19772E 04	0.20530E 04	0.21194E 04	0.21742E 04	0.22175E 04	0.22505E 04	0.22707E 04	0.22799E 04	0.22777E 04	0.22642E 04
0.22398E 04	0.22047E 04	0.21594E 04	0.21044E 04	0.20402E 04	0.19674E 04	0.18655E 04	0.17940E 04	0.17026E 04	0.16007E 04
0.14430E 04	0.13779E 04	0.12522E 04	0.11402E 04	0.10147E 04	0.89623E 03	0.75560E 03	0.60354E 03	0.44057E 03	0.35785E 03
0.22594E 03	0.45434E 02	-0.32562E 02	-0.15748E 03	-0.27462E 03	-0.39537E 03	-0.50719E 03	-0.61350E 03	-0.71421E 03	-0.80695E 03
-0.84573E 03	-0.97814E 03	-0.10527E 04	-0.11202E 04	-0.11365E 04	-0.12335E 04	-0.12792E 04	-0.13172E 04	-0.13475E 04	-0.13701E 04
-0.13848E 04	-0.13917E 04	-0.13909E 04	-0.13824E 04	-0.13555E 04	-0.13437E 04	-0.13140E 04	-0.12780E 04	-0.12352E 04	-0.11990E 04
-0.11370E 04	-0.10437E 04	-0.10229E 04	-0.95755E 03	-0.89162E 03	-0.82340E 03	-0.75344E 03	-0.68194E 03	-0.60925E 03	-0.53600E 03
-0.46244E 03	-0.38687E 03	-0.31599E 03	-0.24393E 03	-0.17325E 03	-0.10433E 03	-0.37620E 02	0.25791E 02	0.85592E 02	0.14360E 03
0.19606E 03	0.24534E 03	0.28977E 03	0.32945E 03	0.35440E 03	0.39461E 03	0.42012E 03	0.44101E 03	0.45741E 03	0.46945E 03
0.47731E 03	0.48112E 03	0.44103E 03	0.47719E 03	0.46973E 03	0.45675E 03	0.44445E 03	0.42634E 03	0.40541E 03	0.38308E 03
0.35722E 03	0.32914E 03	0.29919E 03	0.26777E 03	0.23532E 03	0.20222E 03	0.16513E 03	0.13530E 03	0.10423E 03	0.73332E 02
0.43445E 02	0.16375E 02	-0.91253E 01	-0.32354E 02	-0.53157E 02	-0.71430E 02	-0.87054E 02	-0.99600E 02	-0.11007E 03	-0.11732E 03
-0.12163E 03	-0.12238E 03	-0.12106E 03	-0.11502E 03	-0.10771E 03	-0.96091E 02	-0.81129E 02	-0.62444E 02	-0.41444E 02	-0.16975E 02
0.10336E 02	0.40230E 02	0.72324E 02	0.10532E 03	0.14181E 03	0.17439E 03	0.21559E 03	0.25317E 03	0.29051E 03	0.32762E 03
0.36388E 03	0.39960E 03	0.43274E 03	0.46532E 03	0.49565E 03	0.52434E 03	0.55055E 03	0.57420E 03	0.59503E 03	0.61274E 03

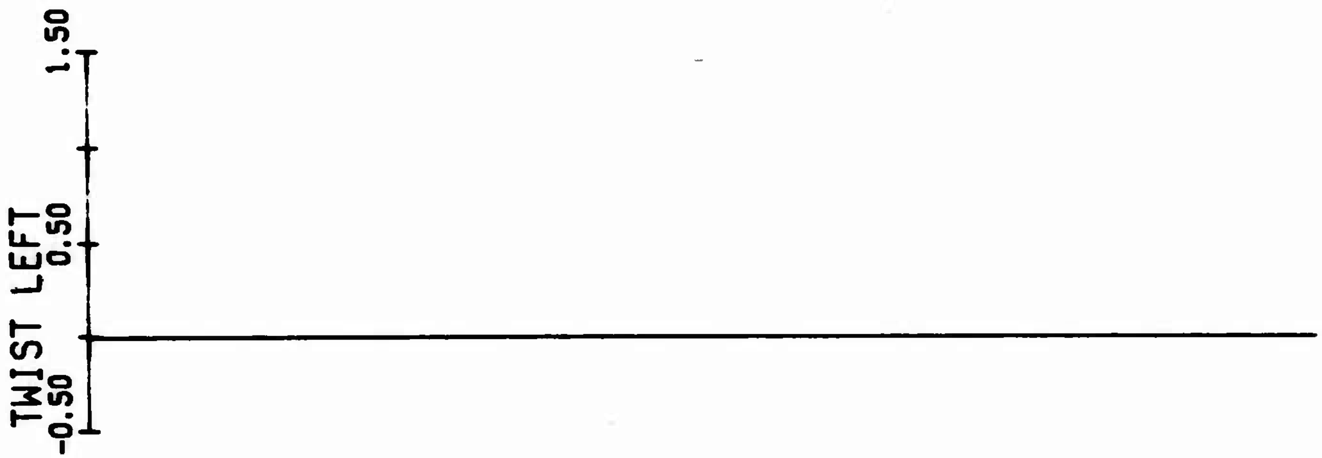
DFAL18 Sample Problem, Condensed Output

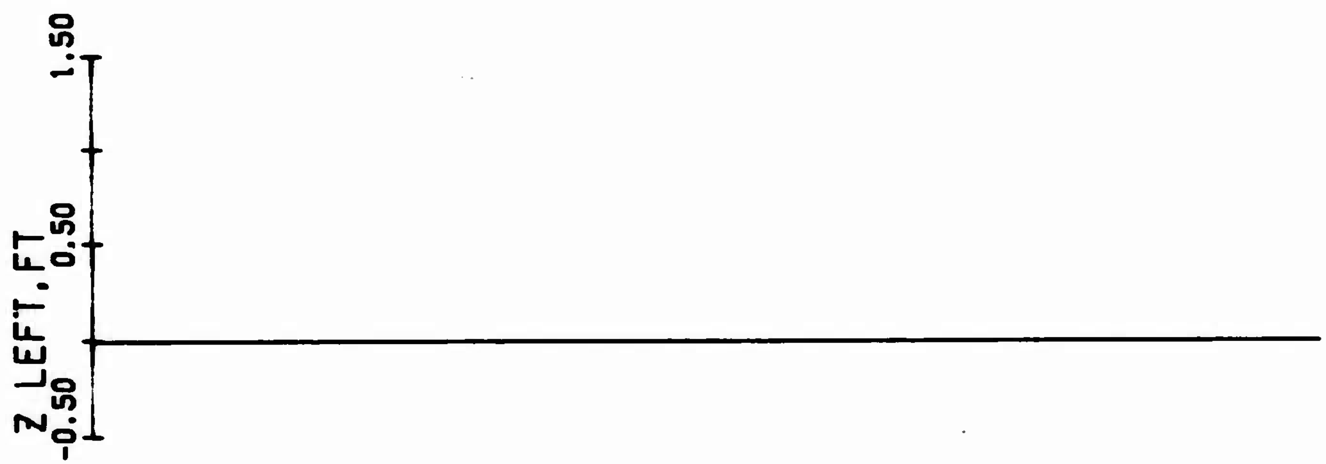


DFAL18 Sample Probler CALCOMP Plots

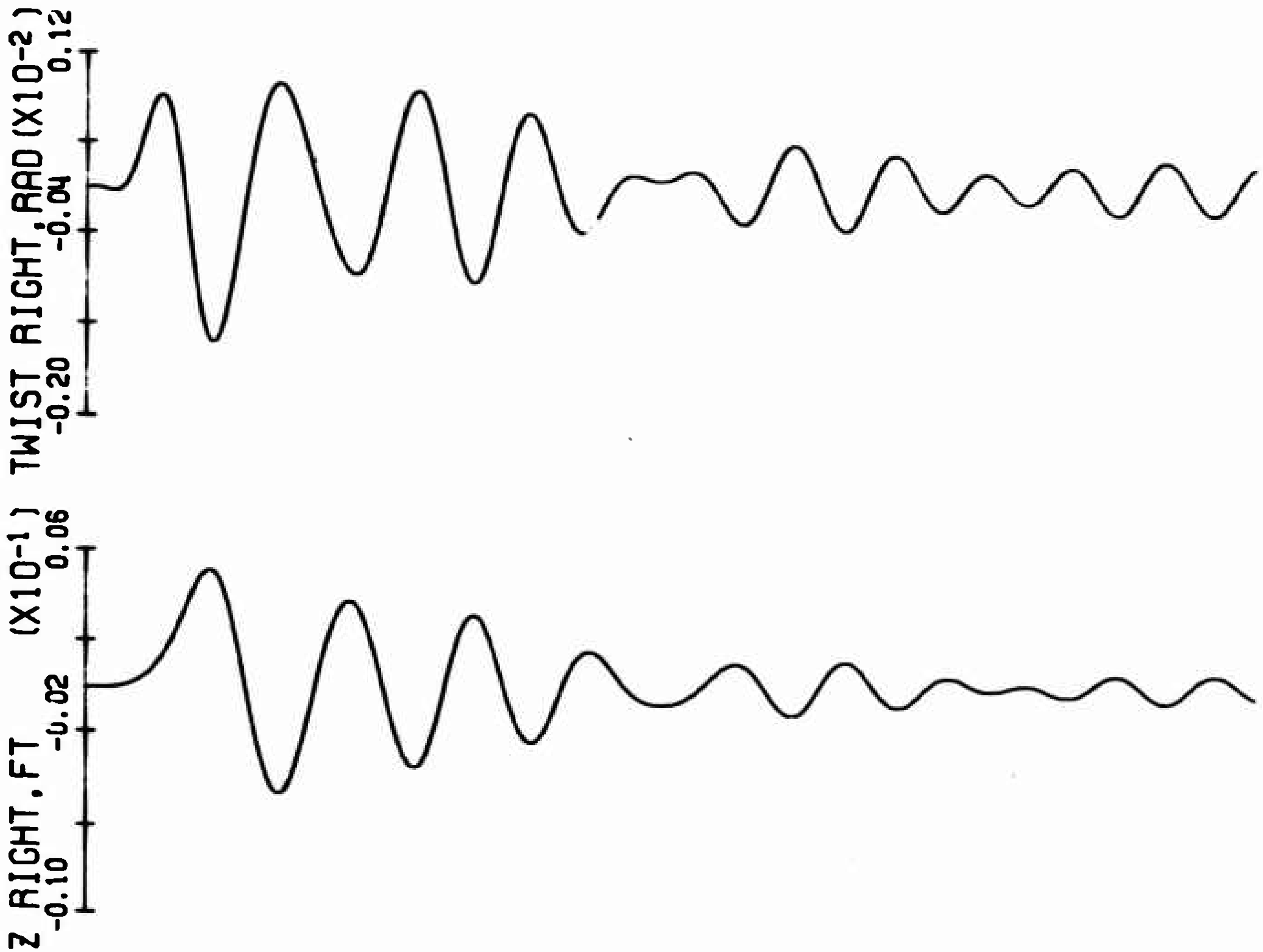


DFAL18 Sample Problem, CALCOMP Plots (Continued)

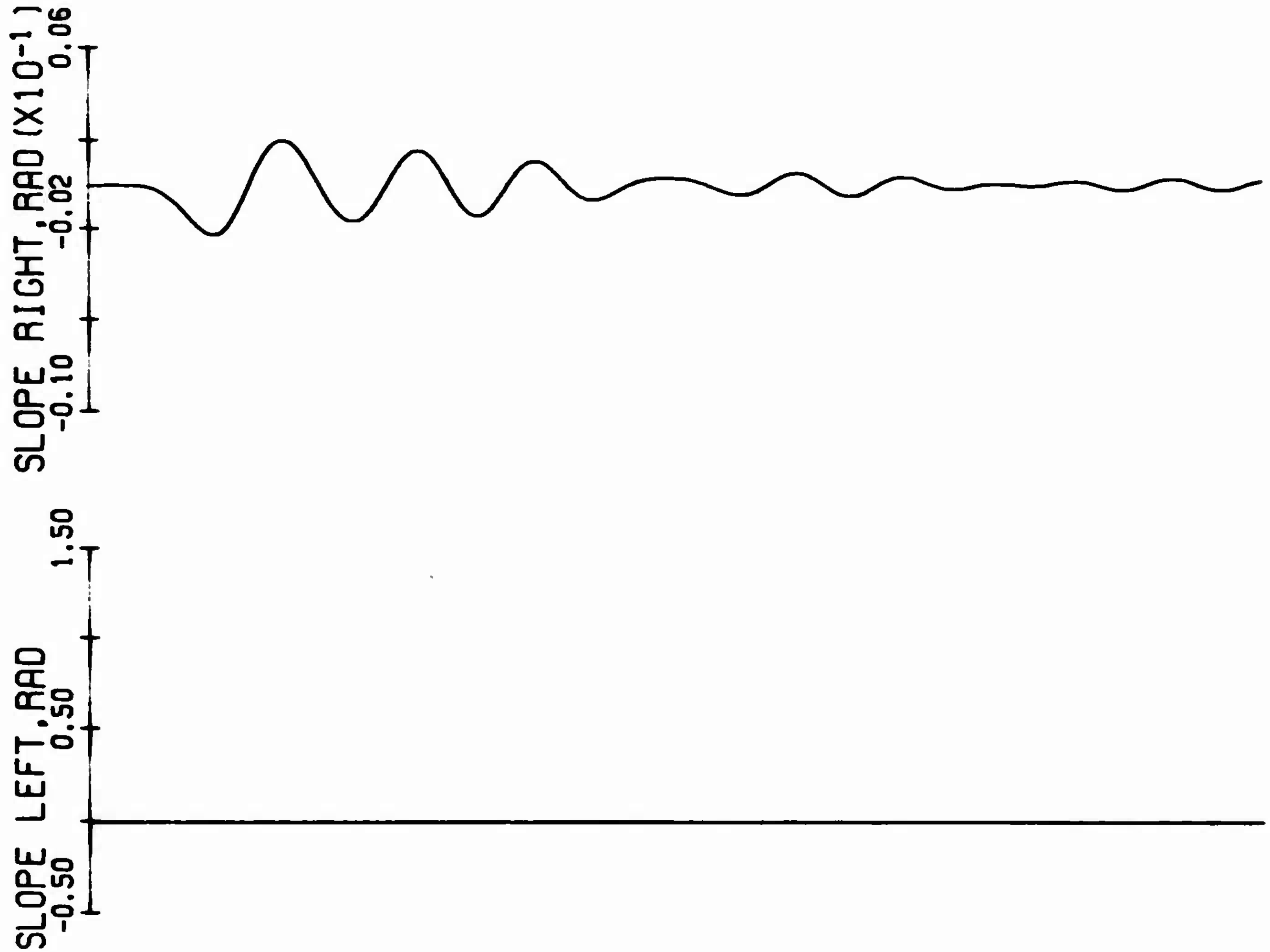




DFAL18 Sample Problem, CALCOMP Plots (Continued)



DFAL18 Sample Problem, CALCOMP Plots (Continued)



DFAL18 Sample Problem, CALCOMP Plots (Concluded)

APPENDIX II

FORTRAN LISTINGS

```
C
C     ARAP06  FOLDING PROPROTOR FEATHER-STOP DYNAMIC ANALYSIS
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON / INPU / AM(10),A0,A1,B1,THCH,DUMK(21),VB(10),
     1  DUM2(20),AIC(10),DUMF(32),
     1  ZH,RHO,DUM7(30),ALL,WC,OMA,CPS,DUM4(2),DTMIN,AMPV,AMPH,THCMAX,
     2  THST,THED,PTHA,PTHB,        DUM1(12),CTHBB(10),STHBB(10),
     2  AIMYS(10),AMY2(10),
     2  CTHT(10),STHT(10),OMAC,OMAS,OMBC,OMBS,A1C,A1S,B1C,B1S,OMACS,
     3  OMASS,OMBCS,OMBSS,TA1B1,SLC,DPSI,GST1,GED1,GST2,GED2,GUST1,GUST2
      COMMON /FLEXOT/ DUME(300),CR
      COMMON /AIRLO/ DUMJ(24),              VINDCA,       CTHP,STHP,SSPH,
     1  CPSIP,CPSI,DUM6(260),DHF(10,3,4),DUM8(90),DT2,DUM5(86),VGH,VGV
      COMMON /FLAP1/ DUML(174),  SCP,SSPP,SPSI,THPDC,STHPP,THPDP,BLG,
     1  BLT,DUMO(20),ATHDDB(3),BTHDDB(3),PSIDDB(3),DUMA(12),A1M,B1M
      COMMON / FLAP4 / CCPH, CSPP, PTP, TSP, CSTP, CCTH,PSID,PSIDD,SPSIP
     1  ,ATH,BTH,PSI,ATHD,BTHD,        ATHDD, BTHDD,DT,AT4DDE,BTHDDE,
     2  PSIDDE,ATHDE,BTHDE,PSIDE,ATHE,BTHE,PSIE,DUM3(3),X2,Y2,XBET(50),
     5  XMBM(50),XWCD(50),XSMP(50),XTBET(50),XTMBM(50),XTWCD(50),
     6  XTSMP(50),DT6,T,TE,PKE(4),CCP
      COMMON /WIN/ DUMC(179),CTPC,STPC
      COMMON /PRIN/ RE(10,4),XI(10,4),YBT(10,4),THM(10,4),DUM9(40),
     2  BEM(4),  BOMSOM(  4),BOMZM2(  4),AMB(10,  4),AMYBT(10,4),
     3  AMX(10,4),AIT(10,4),DUMG(24),BOM(4),BZETA
      COMMON /FLAP6/ IT1,IT2,IT3,IT4,IT5,IT6,IT7,IT8,IT9,IT10,NS,NB,
     1  NRSTN,NGOTO,NRV,JPSI,NOM,NR,NP,KBET,[illegible],KSIP,IDUM1(18),
     2  NOAIR,NC,IDUM2(6),IT14,IBUF(512)
      DATA NCASE/0/
C
   22 CALL INPUT
      VINDCA = CR*24.D0
      IF (RHO.EQ.0.D0 .OR. OMA.EQ.0.D0) VINDCA = 0.D0
      IF (NOAIR.EQ.0) GO TO 26
      DO 30 J=1,NB
      DO 30 I=1,NS
      DO 30 K=1,4
   30 DHF(I,J,K) = 0.D0
   26 NRV = 1
      ATHDDB(1) = 0.
      BTHDDB(1) = 0.
      DT = DTMIN
      DT2 = DT + DT
      PSIDDB(1) = 0.
      DT6 = DT*DT*2./3.
      A1M = A1
      B1M = B1
      TE = 0.D0
      STHP = STPC
      CTHP = CTPC
      SPSIP = 0.D0
      CPSIP = 1.D0
      THPDC = 0.D0
      STHPP = 0.D0
      SCP  = STHP
```

ARAP06 FORTRAN Listings

BELL HELICOPTER COMPANY

```
      THPDP = 0.D0
      TSP = 0.D0
      CCP = CTHP*ZM
      X2 = CCP - ALL
      Y2 = ZM*SCP
      WRITE (6,12)
 12   FORMAT ( 1H1)
      CALL HEADNG
      WRITE (6,24)
 24   FORMAT (//)
      IF (IT6.EQ.0) GO TO 15
      NR = 0
      NP = 0
      CALL PLOTS (IBUF,512)
 15   JPSI = 1
      NRSTN=1
 20   ATHDD = ATHDDB(1)
      BTHDD = BTHDDB(1)
      PSIDD = PSIDDB(1)
      ATHD = ATHDE + (ATHDD + ATHDDE)*DT
      ATH = ATHE + (ATHD + ATHDE)*DT + (ATHDD - ATHDDE)*DT6
      BTHD = BTHDE + (BTHDD + BTHDDE)*DT
      BTH = BTHE + (BTHD + BTHDE)*DT + (BTHDD - BTHDDE)*DT6
      PSID = PSIDE + (PSIDD + PSIDDE)*DT
      PSI = PSIE + (PSID + PSIDE)*DT + (PSIDD - PSIDDE)*DT6
      PSIDS = PSID + PSIDE
      IF (PSIDS.EQ.0.D0) GO TO 40
      DT = DPSI    /DABS(PSIDS)
      DT = DMIN1(DT, DTMIN)
      DT2 = DT+DT
      DT6 = DT*DT*2./3.
 40   T = TE+ DT
      IF (IT14.EQ.0) T = TE + DT2
      SPSI =DSIN(PSI)
      CPSI =DCOS(PSI)
      BLG =      ATH*CPSI - BTH*SPSI
      BLT =      ATH*SPSI + BTH*CPSI
      IF (IT2.EQ.1 .OR. T.LT.TMST .OR. T.GT.TMED) GO TO 44
      THCH = PTHA*T + PTHB
      IF (IT4.EQ.1) GO TO 44
      CALL TABINA(THCH,PSID)
      CT = DCOS(THCH)
      ST = DSIN(THCH)
      DO 54 I=1,NS
      CTHBB(I) = CT*CTHT(I) - ST*STHT(I)
      STHBB(I) = ST*CTHT(I) + CT*STHT(I)
 54   CONTINUE
      DO 52  N=1,NBM
      BEM(  N) = 0.D0
      DO 46  I=1,NS
      BEM(  N) = BEM(  N) + AM(I)*(BE(I,  N)**2 + XI(I,  N)**2) +
     1  AIMYS(I)*THM(I,  N)**2 + AMY2(I)*(CTHBB(I)*THM(I,  N)*BE(I,  N)+
     2  STHBB(I)*THM(I,  N)*XI(I,  N))
      APB(I,  N) = AM(I)*BE(I,  N)
      AMYBT(I,  N) = AM(I)*YB(I)*THM(I,N)
```

ARAP06 FORTRAN Listings (Continued)

```
      AMX(I,  N) = AM(I)*XI(I,  N)
      VBT(I,  N) = VB(I)*THM(I,  N)
      AIT(I,N) = AIC   (I)*THM(I,N)
 46   CONTINUE
      BOMSOM(  N) =              BOM(  N)**2
      BCM2M2(  N) = 2.*BZETA*              BOM(  N)
 52   CONTINUE
 44   IF (IT8.NE.1 .OR. T.LT.TA1B1) GO TO 20
      TT = T-TA1B1
      A1M = A1 + A1C*DCOS((OMAC+TT*OMACS)*TT) + A1S*DSIN((OMAS+TT*OMASS)
     1  *TT)
      B1M = B1 + B1C*DCOS((OMBC+TT*OMBCS)*TT) + B1S*DSIN((OMBS+TT*OMBSS)
     1  *TT)
 20   CALL FLEX1
      IF (IT3.EQ.0) GO TO 68
      GO TO (62,66,64), IT3
 62   IF (T.LT.GST1) GO TO 68
      IF (GST1.NE.GED1) GO TO 42
      VGV = AMPV
      VGH = AMPH
      GO TO 68
 42   IF (T.GT.GED1) GO TO 68
      VGV = (T-GST1)*GUST1*AMPV
      VGH = (T-GST1)*GUST1*AMPH
      GO TO 68
 66   IF (T.LT.GST2.OR. T.GT.GED2) GO TO 68
      GT2 = (DSIN(GUST2*(T-GST2)))**2
      VGV = AMPV*GT2
      VGH = AMPH*GT2
      GO TO 68
 64   SCPST =DSIN(CPS*T)
      VGV = AMPV*SCPST
      VGH = AMPH*SCPST
 68   IF (NOAIR.EQ.0) CALL AIRLDS
      IF (IT4.EQ.0) CALL FLEXBD
      CALL FLAPN1
      CALL FLAPN3
      IF (NGOTO) 15,20,25
 25   IF (IT6.NE.0) CALL PLTS
      IF (IT9.EQ.0) GO TO 48
      IF (KBET.GT.0) WRITE (6,32) (XTBET(K),XBET(K),K=1,KBET)
 32   FORMAT ( ////// 5X,6(5X,'TIME',4X,'BET(1)') // (5X,12F10.3))
      IF (IT5.EQ.0) GO TO 48
      IF (KWBM.GT.0) WRITE (6,34) (XTWBM(K),XWBM(K),K=1,KWBM)
 34   FORMAT ( //// 5X,6(6X,'TIME',7X,'WBM')        // (5X,12F10.3))
      IF (KWCD.GT.0) WRITE (6,36) (XTWCD(K),XWCD(K),K=1,KWCD)
 36   FORMAT ( //// 5X,6(6X,'TIME',7X,'WCD')        // (5X,12F10.3))
      IF (KSHP.GT.0) WRITE (6,38) (XTSHP(K),XSHP(K),K=1,KSHP)
 38   FORMAT ( //// 5X,6(6X,'TIME',4X,'SHP(1)')     // (5X,12F10.3))
 48   NCASE = NCASE + 1
      IF (NCASE.LT.NC) GO TO 22
      IF (IT6.EQ.0) GO TO 50
      CALL PLTIME
      CALL PLOT (5.,3.,999)
 50   STOP
```

ARAP06 FORTRAN Listings (Continued)

```
END                          CARD COUNT     166
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE AIRLDS
      IMPLICIT REAL*8(A-H,O-Z)
      REAL AMM,ALP,ALD,CL,CD,CM,DK,CAD
      COMMON /FLAPW/ IT1,IT2,IT3,IT4,IT5,IT6,IT7,IT8,IT9,IT10,NS,NB,
     1  NRSTN,NGOTO,NRV
      COMMON /INPU/ DUM2(29),R(10),VB(10),EPC(10),EPF(10),AIC(10),
     1  EPF(10),C(10),ANG(10),S,U,
     1  ZH,RHO,RRR(13),DUMA(19),ALW,VCA,UI,VI,TL,R1,R2,R1SQ,R2SQ,ALL,WC
     2  ,DUMF( 9T),SLC
      COMMON/FLAP1/ DUMB(69),TH(10,3),              SBET(3),CBET(3),BETD(3)
     1  ,DUM5(75),SCP,DUMC(3),STHPP,DUM6(14),CPSICP(3),SPSICP(3)
      COMMON /TAIR /               TDCT(10,3),STBSB(10,3),STBCB(10,3),
     1                             CTH(10,3),STH(10,3),         THD(10,3)
      COMMON /FLAPW/ CCPR,CSPP,PTP,DUM(3),PSID,PSIDD,SPSIP,ATM,BTM,
     1  PSI,DUMR(18),Y2,DUMM(407),CCP
      COMMON /AIRLD/ ACCP(3),ASCS(3),ACCS(3),ACSC(3),ASSP(3),
     1           ASSPM(3),ACCPM(3),ACSPP(3),VINDCA,CTHP,
     6  STHP,SSPM,CPSIP,CPSI,MRCDP(10),MRCSDR(10),AVP(10,3),AVT(10,3),
     8  PHI(10,3), AL(10,3), AMACH(10,3), DLIFT(10,3), DDRAG(10,3),
     9  DMOMT(10,3),               DMF(10,3),              DD(10,3),
     X  DRADF(10,3), DTMST(10,3),              CLP(10,3),CDP(10,3),
     1  CMP(10,3),DT2,PH(10,3),PHE(10,3),BB(10),TOC(10),CPSIA(3),
     2  SPSIA(3),VGH,VGV
      COMMON /WINGT/ XOD,YOD,ZOD,THPD,PSIPD
      COMMON /FLEXOT/ FT(10,3),ETD(10,3),ETDD(10,3),ETM(10,3),ETMD(10,3)
     1  ,ETMDD(10,3),ETV(10,3),ETVD(10,3),ETVDD(10,3),MPETSQ(10,3),CB,SB
      DIMENSION                    FR1(10,3), FR2(10,3), UVP1(3),CA(3),
     1  UVP2(3), UVT1(3) ,UVT2(3),BVP(3),BVT(3),NSTL(6),NEDL(6),PPSI(6)
      DATA                         PPSI / 1.0472,2.0944,3.1416,4.1888,
     1  5.236,6.2832/,NSTL,NEDL/3,2,2,3*1,3,3,2,2,1,3/
C
      DO 200 J=1,NB
      CA(J) = CBET(J)*CB + SBET(J)*SB
      UVP1(J) = U*(CTHP*ACCP(J) + ASCS(J))
      UVP2(J) = U*(-ACCS(J) + STHP*ACCP(J))
      UVT1(J) = U*ACSC(J)
      UVT2(J) = U*ASSPM(J)
      ASCP = SPSIP*CBET(J) + SPSICP(J)*SBET(J)
      BVP(J)=-XOD*(CTHP*ACCP(J)+ASCS(J))+YOD*ASCP   +ZOD*(ACCS(J)-STHP*
     1  ACCP(J))+ZH*ACCPM(J)*SBET(J)           -VINDCA
      BVT(J) = XOD*ACSC(J) +  YOD*CPSICP(J) - ZOD*ASSPM(J) - ZH*ACSPP(J)
      DO 200 I=1,NS
      FR1(I,J) = 1.D0
  200 FR2(I,J) = 0.D0
      IF (IT3.EQ.0) GO TO 50
      DO 35 J=1,NB
      BVP(J) = BVP(J) - VGH*(ASCS(J) + CTHP*ACCP(J)) - VGV*(ACCS(J) -
     1  STHP*ACCP(J))
   35 BVT(J) = BVT(J) + VGH*ACSC(J) + VGV*ASSPM(J)
   50 IF (IT7.EQ.0) GO TO 70
      NIC = 1
      DO 22 K=1,5,2
      KK = K
      IF (PSI.GT.PPSI(K+1)) GO TO 22
      IF (PSI.LT.PPSI(K)) GO TO 24
```

ARAP06 FORTRAN Listings

BELL HELICOPTER COMPANY

```
      KK = K+1
      GO TO 24
22    CONTINUE
      KK = 1
24    NST = NSTL(KK)
      NED = NEDL(KK)
      IF (KK.EQ.6) NIC = 2
      DO 85 J=NST,NED,NIC
      XY1     = CBET(J)*SPSIA(J)*SPSIP + SBET(J)*CPSIP
      X1      = CBET(J)*CPSIA(J)*STHP - XY1    *CTHP
      Y1      = CBET(J)*CPSIA(J)*CTHP + XY1    *STHP
      XY01 = SPSIA(J)*CB*SPSIP + SB*CPSIP
      XY02 = SPSIA(J)*SB*SPSIP - CB*CPSIP
      X01 = CPSIA(J)*CB*STHP - XY01*CTHP
      X02 = CB*CPSIP*CTHP + (CPSIA(J)*STHP - SPSIA(J)*SPSIP*CTHP)*SB
      X03 =-SB*SPSIP + CB*SPSICP(J)
      X04 = CB*SPSIP + SB*SPSICP(J)
      X05 = STHP *XY01 + CB*CPSIA(J)*CTHP
      X06 = SB*CPSIA(J)*CTHP + XY02*STHP
      DO 85 I=1,NS
      XY2        = SPSIA(J)*CTH (I,J) + CPSIA(J)*STBSB(I,J)
      XY3        = CPSIA(J)*CTH (I,J) - STBSB(I,J)          *SPSIA(J)
      X3       = STHP*XY2         *CTHP*STBCB(I,J)*CPSIP + XY3*    SPSIP*CTHP
      E1 = CCP - R(I)*X1 + HRETSO(I,J)*X01 + ET(I,J)*X02 - FTN(I,J)*
     1  ACSC(J) + VR(I)*X3
      E2 = ZH*SPSIP - R(I)*ASSP(J) - HRETSO(I,J)*X03 - ET(I,J)*X04 -
     1  FTN(I,J)*CPSICP(J) + VR(I)*(CPSIP*XY3 - STBCB(I,J)*SPSIP)
      E3 = HRETSO(I,J)*X05 - Y2 - R(I)*Y1 + ET(I,J)*X06 - ETN(I,J)*
     1  ASSPH(J) + (XY2*CTHP - STBCB(I,J)*SCP - XY3*STHPP)*VR(I)
      XI = (E3*VI - E1*UI + ALL + (E2+VCA)*TL)/WC - .5D0
      Y = -(E1*VI + E3*UI)/WC
      YSQ = Y*Y
      ER = XI*XI*.25 - YSQ*.25 - R2SQ
      EI = XI*Y*.5
      RR =DSQRT(ER*ER + EI*EI)
      SN = 1.D0
      IF (XI.LT.0.D0) SN = -1.D0
      XZ = DSQRT((RR + ER)*.5D0)*SN + XI*.5D0
      YZ = DSQRT((RR - ER)*.5D0)*DSIGN(1.D0,EI)*SN + Y*.5D0
      XZSQ = XZ*XZ
      YZSQ = YZ*YZ
      AB1 = ((XZ+R2)**2 + YZSQ)*((XZ-R2)**2 + YZSQ)
      DUE = ((XZSQ + YZSQ)**2 + R1SQ*R2SQ - (XZSQ-YZSQ)*(R1SQ+R2SQ))/AB1
      DVE = -2.*YZ*XZ*(R1SQ - R2SQ)/AB1
      DUFP = 0.D0
      DVFP = 0.D0
      IF (ALW.EQ.0.D0) GO TO 140
      D1 = XI - .5
      D2 = XI + .5
      D3 = XI - SLC/1.2566370D1
      PHI2 = DATAN2(Y,D3) - (DATAN2(Y,D1) + DATAN2(Y,D2))*.5D0
      R12 =DSQRT((D3*D3 + YSQ)/DSQRT((D1*D1 + YSQ)*(D2*D2 + YSQ)))
      DUFP = VI*R12*DSIN(PHI2)
      DVFP = VI* R12*DCOS(PHI2) - VI
140   VRL = UI*DUE + DUFP
```

ARAP06 FORTRAN Listings (Continued)

```
      VIM = VI + DVFP + DVE*UI
      FR1(I,J) = VRL*UI + VIM*VI
      FR2(I,J) = VIM*UI - VRL*VI
   85 CONTINUE
   79 DO 75 J=1,NB
      FB131 = PTP*SBET(J) + CBET(J)*ACSPP(J)
      F1    = SB*PTP + CB*ACSPP(J)
      F2 = PTP*CBET(J) - ACSPP(J)*SBET(J)
      F3 = PTP*CB        - ACSPP(J)*SB
      APCC = THPD*ASSP(J) - PSIPD*CPSIA(J)*CBET(J) - PSID*SBET(J)
      DO 75 I=1,NS
      RLD = RRR(I)
      AVP(I,J) =      -UVP1(J)*FR1(I,J) + UVP2(J)*FR2(I,J)
     X    + BVP(J) + (ACCPM(J) - BETD(J))*R(I) + VB(I)*CT4(I,J)*(APCC
     1           - THD(I,J)              ) + EPC(I)*TDCT(I,J) + (ETM(I,J)*
     2  F1     - ETD(I,J) - HRETSQ(I,J)*ACCPM(J))*CA(J)
      AVT(I,J) =        UVT1(J)*FR1(I,J) + UVT2(J)*FR2(I,J)
     1    + BVT(J) + F2*R(I) - F3*HRETSQ(I,J) - VB(I)*(THD(I,J) + FB131)*
     2  STH(I,J) + EPC(I)*THD(I,J)*STH(I,J)- ET(I,J)*F1    - ETMD(I,J)
      AVSQ        = AVP(I,J)**2 + AVT(I,J)**2
      AV        =DSQRT(AVSQ       )
      PH(I,J) = DATAN2(AVP(I,J),AVT(I,J))
      ALP = ((C(I)*.5D0 - EPF(I))*THD(I,J)/AV + TH(I,J) +PH(I,J)
     1  + ETV(I,J))*57.296
      AMM = AV/S
      DK = TOC(I)*AV
      CALL CLCDM (AMM,ALP,RLD   ,CL,CD,CM,     DK,CAD)
      DLIFT(I,J) = CL*HPCDR(I)*AVSQ
      DDRAG(I,J) = CD*HPCDR(I)*AVSQ
      DMOMT(I,J) = CM*HPCSDR(I)*AVSQ
      DTTPP      = (DLIFT(I,J)*AVT(I,J) + DDRAG(I,J)*AVP(I,J)) / AV
      DHF(I,J) = (DDRAG(I,J)*AVT(I,J) - DLIFT(I,J)*AVP(I,J)) / AV
      DQ(I,J) = (CTH(I,J)*DTTPP + STH(I,J)*DHF(I,J))*EPE(I) + DMOMT(I,J)
     1  + ((PH(I,J) - PHE(I,J))/DT2 + THD(I,J))*BB(I)*CAD
      DRADF(I,J) = DTTPP      *SBET(J)
      DTMST(I,J) = DTTPP      *CBET(J)
      IF (NPV.NE.2 .AND. PSID.NE.0.D0) GO TO 75
      PHI(I,J) = 57.296*PH(I,J)
      AMACH(I,J) = AMM
      AL (I,J) = ALP
      CLP(I,J) = CL
      CDP(I,J) = CD
      CMP(I,J) = CM
   75 CONTINUE
      RETURN
      END
                             CARD COUNT      156
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE BEXION
      REAL*8 DUM1,ZEE
      COMMON /FRIN/ DUM1(196),ZEE(3,4)
      COMMON /FRIN1/ FTHC(6),FPSID(6),OMG(6,6,4),DFTHC(5),DFPSID(5),
     1  FBE(1440),FXI(1440),FTH(1440),BMT(6,6,4,4),CMT(6,6,4,4)
      COMMON /FLAP6/ IDUM1(10),NS,NB,IDUM2(4),NBM,IDUM3(10),NTHC,
     1  NPSID,NMT,IDUM4(13),KOR(4)
      DIMENSION FBXT(10,6,6,4,3),LC(3),NFT(10),NOMG(7),KLC1(32),
     1  KLC2(26),KLC3(26),FTT(20)
      DATA NFT/'    1    2    3    4    5    6    7    8    9   10'/,  NOMG/'(/(10
     1X,F10.1,2X,   9F10.1)) '/,      KLC1/'(/T10,3HLC(,I1,3H) =,I2,(/T10
     2 ,5HRPM =,F5.1,8H RAD/SEC/T21,7HSTATION,I3,    8I10 /  T17,3HTHC,
     3     5(/ 10X,F10.1,2X,    5F10.3)))'/,      KLC2/'(/T10,3HLC(,I1, 3H)
     4 =,I2,2X,11HFOR ALL RPM/T21,7HSTATION,I3,    9I10/T17,3HTHC/(10X,F1
     50.1,2X,    5F10.3)) '/,      KLC3/'(/T10,3HLC(,I1,3H) =,I2,2X,11HFOR
     6 ALL THC/ T21,7HSTATION,I3,    9I10/T17,3HRPM/(10X,F10.1,2X,    5F10
     7.3)) '/
      EQUIVALENCE (FBXT(1,1,1,1,1),FBE(1)),(FBXT(1,1,1,1,2),FXI(1)),
     1  (FBXT(1,1,1,1,3),FTH(1))
C
   10 FORMAT (16I5)
   16 FORMAT (13F6.0,2X)
   18 FORMAT (9F8.0,8X)
   20 FORMAT (20A4)
   22 FORMAT (// T10,'BEAM MOMENT AT SPECIFIED STATION',I2 // T15,
     1  'THC (DEG)', T40,'ROTOR RPM (RAD/SEC)' /  22X,5F10.1)
   24 FORMAT (// T10,'CHORD MOMENT AT SPECIFIED STATION',I2 // T15,
     1  'THC (DEG)', T40,'ROTOR RPM (RAD/SEC)' /  22X,6F10.1)
   28 FORMAT (/ ' ZE(J,M)',17X,15F7.3)
   30 FORMAT (/// ' PROPROTOR BLADE NORMAL MODES')
   34 FORMAT (/ 1X,20A4 // ' KOR(',I1,') =',I2 // T10,
     1  'NATURAL FREQUENCY' // T15,'THC (DEG)', T40,'ROTOR RPM (RAD/SEC)
     2' / 22X,6F10.1)
   62 FORMAT ( //T10,'OUT-OF-PLANE MODE SHAPE')
   64 FORMAT ( //T10,'INPLANE MODE SHAPE')
   65 FORMAT ( //T10,'TORSIONAL MODE SHAPE')
   67 FORMAT ( /T10,'LC(',I1,3H) =,I2, T21,'FOR ALL THC AND RPM' / T21,
     1  'STATION',I3, 9I10)
   78 FORMAT (/ 22X,10F10.3)
C
      NOMG(5) = NFT(NPSID)
      KLC1(19) = NFT(NS-1)
      KLC1(25) = NFT(NTHC)
      KLC1(30) = NFT(NS)
      KLC2(16) = NFT(NS-1)
      KLC2(24) = NFT(NS)
      KLC3(16) = NFT(NS-1)
      KLC3(24) = NFT(NS)
      WRITE (6,30)
      READ (5,16) (FTHC(N),N=1,NTHC),(FPSID(N),N=1,NPSID)
      DO 52 N=1,NBM
      READ (5,20) FTT
      READ (5,16) ((OMG(M,K,N),M=1,NTHC),K=1,NPSID)
      READ (5,17) (LC(L),L=1,3),KOR(N)
      WRITE (6,34) FTT,N,KOR(N),(FPSID(M),M=1,NPSID)
```

ARAP06 FORTRAN Listings (Continued)

```
      WRITE (6,NOMG) (FTHC(K),(OMG(K,M,N),M=1,NPSID),K=1,NTHC)
      DO 100 NBXT=1,3
      LCN = LC(NBXT)
      M2 = NTHC
      K2 = NPSID
      IF (LCN.GT.2) M2 = 1
      IF (LCN.FO.2 .OR. LCN.EQ.4) K2 = 1
      READ (5,16) (((FBXT(I,M,K,N,NBXT),I=1,NS),M=1,M2),K=1,K2)
      IF (NBXT.FO.1) WRITE (6,62)
      IF (NBXT.FO.2) WRITE (6,64)
      IF (NBXT.FO.3) WRITE (6,65)
      GO TO (68,70,72,74), LCN
68    WRITF (6,KLC1) NBXT,LCN,(FPSID(K),(II,II=1,NS),(FTHC(M),
     1  (FBXT(I,M,K,N,NBXT),I=1,NS),M=1,M2),K=1,K2)
      GO TO 76
70    WRITE (6,KLC2) NBXT,LCN,(I,I=1,NS),
     1               (FTHC(M),(FBXT(I,M,1,N,NBXT),I=1,NS),M=1,M2)
      GO TO 76
72    WRITE (6,KLC3) NBXT,LCN,(I,I=1,NS),
     1               (FPSID(K),(FBXT(I,1,K,N,NBXT),I=1,NS),K=1,K2)
      GO TO 76
74    WRITE (6,67) NBXT,LCN,(I,I=1,NS)
      WRITE (6,78) (FBXT(I,1,1,N,NBXT),I=1,NS)
76    IF (M2.FO.NTHC .AND. K2.EO.NPSID) GO TO 100
      M1 = 1
      K1 = 1
      DC 27 M=1,NTHC
      IF (M2.EQ.NTHC) M1 = M
      DO 27 K=1,NPSID
      IF (K2.EQ.NPSID) K1 = K
      DO 27 I=1,NS
27    FBXT(I,M,K,N,NBXT) = FBXT(I,M1,K1,N,NBXT)
100   CONTINUE
      DO 52 L=1,NMT
      READ (5,19) ((BMT(M,K,L,N),M=1,NTHC),K=1,NPSID)
      WRITE (6,22) L,(FPSID(K),K=1,NPSID)
      WRITE (6,NOMG)   (FTHC(K),(BMT(K,M,L,N),M=1,NPSID),K=1,NTHC)
      READ (5,19) ((CMT(M,K,L,N),M=1,NTHC),K=1,NPSID)
      WRITE (6,24) L,(FPSID(K),K=1,NPSID)
      WRITE (6,NOMG)   (FTHC(K),(CMT(K,M,L,N),M=1,NPSID),K=1,NTHC)
52    CONTINUE
      NTHC1 = NTHC-1
      DO 50 M=1,NTHC1
50    DFTHC(N) = FTHC(N+1) - FTHC(N)
      NPSID1 = NPSID-1
      DO 55 N=1,NPSID1
55    DFPSID(N) = FPSID(N+1) - FPSID(N)
      READ (5,16) ((ZEE(J,N),J=1,NB),N=1,NBM)
      WRITF (6,20) ((ZEE(J,N),J=1,NB),N=1,NBM)
      RETURN
      END
                              CARD COUNT       106
```

ARAP06 FORTRAN Listings (Continued)

PAGE 1

```
      BLOCK DATA
      COMMON /LLDD/ DUM(2),CURVE(2688)
      DIMENSION    CM3B(140),CM4R( 90),CL3(395),CALD(159),
     1  CD3(395),CL4( 311),CD4( 311),CL3B(220),CD3B(207),CL4B(161),
     2  CD4B(125),CM2(179),              CM3(340),CM4(285),CL2(161),CD2(152)
      EQUIVALENCE (CL3B(1),CL3(176)),(CD3B(1),CD3(196)),
     1  (CL4B(1),CL4(151)),(CD4B(1),CD4(187)),
     2  (CM3B(1),CM3(201)), (CM4B(1),CM4(196)), (CURVE(493),CL3(1)),
     3  (CURVE( 888),CD3(1)),(CURVE(1283),CM3(1)),(CURVE(1623),CL4(1)),
     4  (CURVE(1934),CD4(1)),(CURVE(2245),CM4(1)),(CURVE(2530),CALD(1)),
     1  (CURVE(162),CD2(1)),(CURVE(314),CM2(1)),(CURVE(1),CL2(1))
      DATA CL2  / 0. ,  .28 ,  .3  ,  .4  ,  .5  ,  .61 ,  .66 ,   1.
     1 ,-180. ,  -8. ,-6.   ,-4.   ,-2.   , 0.  , 2.  , 4.  ,   6.
     2 ,   8. ,  10. ,12.   ,14.   ,16.   ,18.  ,20.  , 100.
     3 , 8* .0, -.858, -.858, -.86 , -.875, -.85 , -.855, -.78 ,  0.
     4 , 3*-.625      , -.645, -.66 , -.695, -.57 ,  .0  , 3*-.395
     5 , -.405, -.415, -.4  , -.355,  .0  , -.17 , -.17 , -.165,  -.17
     6 , -.19 , -.19 , -.15 ,  .0  , 4*.05, 2*.06,  .035,  .0  , 3* .27
     7 ,  .275,  .27 ,  .295,  .22 ,  .0  ,2*.405,  .48 ,2*.495,  .47
     8 ,  .41 ,  .0  ,2*.69 ,  .685,  .705,  .71 ,  .675,  .61 ,  0.
     9 ,2*.88 ,  .885,  .91 ,  .915,  .895,  .8  ,  .0  ,2*1.08,  1.065
     X , 1.97 ,  .97 , 1.015,  .905,  .0  , 2*1.165      , 1.175,  1.105
     1 , 1.   , 1.075,  .96 ,  .0  , 2*1.165       , 1.17 , 1.065,  1.
     2 , 1.03 ,  .995,  .0  ,3*1.14, 1.005, 1.    ,  .91 , 1.   ,  0.
     3 ,2*1.09, 1.135,  .905,  .99 ,  .815,  .9   ,  .0  , 2*1.025
     4 , 1.265,  .78 ,  .98 ,  .43 ,  .8  ,  .0  ,8*.0  /
      DATA CD2  / 0. ,  .26 ,  .3  ,  .4  ,  .5  ,  .61 ,  .66 ,   1.
     1 ,-180. ,  -8. ,-6.   ,-4.   ,-2.   ,        2.  , 4.  ,   6.
     2 ,   8. ,  10. ,12.   ,14.   ,16.   ,18.  ,20.  , 180.
     3 , 8*0. ,2*.014,2*.013,  .016,  .022,  .025, 0.  ,3*.011,   .01
     4 ,  .011,  .012,  .01 ,  .0  ,2*.009,  .01 ,3*.009,  .008,  0.
     5 ,7*.007,  .0  ,         7*.007,  .9  ,3*.007,  .008, 2* .007
     6 ,  .008,  .0  ,2*.01 ,  .009,2*.01 ,  .007,  .013,  .0  , 2* .014
     7 ,2*.013,2*.014,  .037,  .0  ,3*.02 ,  .033,  .054,  .055,   .002
     8 ,  .0  ,2*.071,  .066,  .075,  .083,  .068,  .19 ,  .0  , 3* .101
     9 ,  .104,  .124,  .213,  .25 ,  .0  ,3*.127,  .14 ,  .216,   .307
     X ,  .344,  .0  ,3*.182,  .216,  .304,  .406,  .441,  .0  , 2* .017
     1 ,  .02 ,  .01 ,  .0  ,  .006, 10* .0            /
      DATA CM2  / 0.., .251,        .402, .503, .604, .656, .71, .772, 1..
     1   -180.,-8.,-6.,-4.,-2.,0.,2.,4.,6.,8.,10.,12.,14.,16.,18.,20..
     4   180.,  2*-.01, -.014, -.017, -.007, -.009, -.002, -.108, 0..
     7   2*-.01, -.011, -.014, -.007, -.009, -.002, -.108, 0..
     9   4*-.009,            -.007, -.009, -.002, -.108, 0..
     X   2*-.009, 2*-.008, 2*-.007, -.009, -.111, 0.,  3*-.008, -.009,
     1   -.021, -.076, 0..  7*-.008, .007, 0..  3*-.009, -.01, -.013,
     2   -.012, -.009, .048, 0..  3*-.01, -.012, 2*-.014, -.018, .075,
     3   0..  3*-.011, -.013, -.014, -.013, -.026, .063, 3.., 2*-.009,
     4   3*-.008, -.019, -.034, .065, 0..
     1   2*-.036, -.008, -.011, -.004, -.034, -.063, .065, 0.., 2*-.016,
     3   -.015, .005, -.013, -.054, -.067, .065, 0.., 2*-.022,-.018,.001,
     5   -.056, -.081, -.067, .065, 0.., 2*-.026,           -.028, -.052,
     7   -.056, -.097, -.067, .065, 0.., 2*-.026,           -.028, -.092,
     8   -.056, -.097, -.067, .065, 0.., 2*-.026,           -.028, -.092,
     9   -.056, -.097, -.067, .065, 0.., 2*-.01 ,           -.014, -.017,
     X   -.007, -.009, -.002, -.108, 0.  /
```

ARAP06 FORTRAN Listings (Continued)

```
      DATA CL3  / 0. ,  .3  ,  .4  ,  .5  ,  .6  ,  .66 ,  .71 ,   .76
     1 ,  .82 ,  .85 , 1.   ,-180. , -10. ,  -9. ,  -8. ,  -7. ,  -6.
     2 ,  -5. ,  -4. ,  -3. ,  -2. ,  -1. ,   0. ,   1. ,   2. ,   3.
     3 ,   4. ,   5. ,   6. ,   7. ,   8. ,   9. ,  10. ,  11. ,  12.
     4 ,  13. ,  14. ,  15. ,  16. ,  17. ,         19. ,  20. , 180.
     1                    ,11*0. ,      2*-.767, -.741, -.706,   -.499
     2 , -.88 , -.87 ,4*.0  ,2*-.75, -.72 , -.705, -.64 , -.865,   -.85
     3 , -.8  ,3*.0  ,3* -.708      , -.7  , -.72 , -.833, -.818,   -.772
     4 , -.71 , -.55 ,  .0  ,3*-.65, -.69 , -.7  , -.765, -.74 ,   -.68
     5 , -.655, -.495,  .0  ,2*-.559       , -.589, -.587, -.626,   -.656
     6 , -.659, -.595, -.585, -.44 ,  .0  ,2*-.45, -.47 , -.5  , 2*-.54
     7 , -.55 , -.51 , -.48 , -.37 ,  .0  ,2*-.343        , -.357,   -.38
     8 , -.405, -.414, -.425, -.396, -.375, -.31 ,  .0  ,2*-.25,   -.24
     9 ,2*-.27, -.29 , -.3  , -.28 , -.265, -.24 ,  .0  ,2*-.123
     X , -.125, -.132, -.146, -.155, -.14 , -.144, -.16 , -.155,    .0
     1 ,2*-.03, -.01 , -.03 , -.031, -.032, -.1  , -.04 ,2*-.06,    .0
     2 ,2*.078,  .085,  .089,  .093,2*.099,  .072,  .059,  .05 ,    .0 /
      DATA CL38 /
     3  2*.18 ,  .21 ,  .213,  .233,  .24 ,  .23 ,  .205,  .18 ,    .19
     4 ,  .0  ,2*.339,  .318,  .338,  .363,  .365,  .362,  .315,    .27
     5 ,  .228,  .0  ,2*.4  ,  .439,  .467,  .502,  .485,  .48 ,    .425
     6 ,  .39 ,  .315,  .0  ,2*.523,  .551,  .583,  .627,  .623,    .604
     7 ,  .537,  .485,  .41 ,  .0  ,2*.62 ,  .665,  .68 ,  .735,    .73
     8 ,  .7  ,  .62 ,  .595,  .5  ,  .0  ,2*.735,  .781,  .825,    .883
     9 ,  .874,  .797,  .709,  .66 ,  .6  ,  .0  ,2*.83 ,  .885,    .92
     X ,  .96 ,  .94 ,  .872,  .76 ,  .74 ,  .69 ,  .3  ,2*.956,   1.012
     1 , 1.02 , 1.054, 1.005,  .948,  .82 ,  .81 ,  .78 ,  .0  , 2*1.05
     2 , 1.088, 1.083, 1.11 , 1.05 , 1.01 ,  .859,2*.88 ,  .0  , 2*1.159
     3 , 1.125, 1.095, 1.147, 1.085, 1.057,  .899,  .93 ,  .96 ,    .0
     4 ,      2* 1.296, 1.11 , 1.048, 1.095, 1.06 , 1.105,4*.0  , 2*1.254
     5 , 1.086, 1.005, 1.044, 1.046, 1.142,4*.0  ,2*1.15, 1.04 ,    .96
     6 ,  .99 , 1.075, 1.1  ,4*.0  ,2*.945,  .977,  .91 ,  .958,   1.066
     7 , 1.045,4*.0  ,2*.84 ,  .89 ,8*.0  ,2*.8  ,  .803,8*.0  , 2* .78
     8 ,9*.0  ,             2*.782,9*.0  ,2*.786,9*.0  ,11*.0 /
      DATA CD3  / 0. ,  .3  ,  .4  ,  .5  ,  .6  ,  .66 ,  .71 ,   .76
     1 ,  .82 ,  .85 , 1.   ,-180. , -10. ,  -9. ,  -8. ,  -7. ,  -6.
     2 ,  -5. ,  -4. ,  -3. ,  -2. ,  -1. ,   0. ,   1. ,   2. ,   3.
     3 ,   4. ,   5. ,   6. ,   7. ,   8. ,   9. ,  10. ,  11. ,  12.
     4 ,  13. ,  14. ,  15. ,  16. ,  17. ,         19. ,  20. , 180.
     5 ,11*0. ,3*.132,  .135,  .153,  .184,  .231,4*.0  ,2*.101,    .1
     6 ,  .105,  .112,  .128,  .173,  .232,3*.0  ,2*.059,  .068,    .08
     7 ,  .058,  .066,  .098,  .143,  .23 ,  .259,  .0  ,2*.032,    .04
     8 ,  .053,  .05 ,  .047,  .07 ,  .107,  .196,  .232,  .0  , 2* .01
     9 ,  .012,  .021,  .022,  .028,  .041,  .066,  .17 ,  .208,    .0
     X ,3*.01 ,  .015,2*.013,  .014,  .037,  .135,  .193,  .0  , 2* .01
     1 ,2*.009,  .008,  .01 ,  .011,  .014,  .118,  .177,  .0  , 2* .009
     2 ,2*.008,  .007,2*.009,  .01 ,  .105,  .163,  .0  ,4*.008,    .007
     3 ,2*.008,  .009,  .097,  .149,  .0  ,4*.007,2*.006,  .007,    .01
     4 ,  .095,  .149,  .0  ,7*.006,  .013,  .091,  .149,  .0  , 5* .006
     5 ,2*.007,  .019,  .108,  .16 ,  .0  ,4*.006,  .007,  .006,    .008 /
      DATA CD38 /
     6    .035,  .128,  .175,  .0  ,2*.008,2*.007,2*.006,  .01 ,    .058
     7 ,  .152,  .195,  .0  ,2*.009,3*.008,  .007,  .016,  .06 ,    .175
     8 ,  .218,  .0  ,2*.01 ,  .008,  .009,2*.01 ,  .023,  .085,    .198
     9 ,  .235,  .0  ,4*.01 ,  .011,  .013,  .035,  .095,  .215,    .253
```

ARAP06-FORTRAN Listings (Continued

```
     X ,  .0  ,2*.011,  .012,  .015,  .019,  .02 ,  .059,  .139,    .238
     1 ,  .275,  .0  ,3*.013,  .019,  .027,  .035,  .088,  .168,    .261
     2 ,  .299,  .0  ,2*.014,  .018,  .035,  .048,  .064,  .131,    .2
     3 ,  .287,  .316,  .0  ,2*.017,  .036,  .06 ,  .056,  .093,    .161
     4 ,  .23 ,  .335,  .339,  .0  ,2*.023,  .11 ,  .098,  .123,    .135
     5 ,  .205,4*.0  ,2*.034,  .132,  .136,  .168,  .187,  .224, 4* .0
     6 ,2*.137,  .186,  .191,  .2  ,  .195,  .243,4*.0  ,3*.211,    .218
     7 ,  .229,  .235,  .266,4*.0  ,2*.23 ,  .245,8*.0  ,2*.248,    .253
     8 ,8*.0  ,2*.273,9*.0  ,            2*.298,9*.0  ,2*.32 , 9* .0
     9 ,11*.0 /
      DATA CM3  / 0., .391, .401, .502, .603, .654, .705, .758, .795, 1.
     1  , -180., -19., -9., -8., -7., -6., -5., -4., -3., -2., -1., 0.,
     2  1., 2., 3., 4., 5., 6., 7., 8., 9., 10., 11., 12., 13., 14., 15.
     3  , 16., 20., 180.,
     4  2*.045, .047, .057, -.045, -.028, -.007, .048, .037, 0., 2*.045,
     5  .047, .057, -.045, -.028, -.007, .048, .037, 0., 2*.007, .005,
     6  .021, -.045, -.028, -.007, .048, .037, 0., 2*-.019, -.015, -.006
     7  , -.045, -.028, -.007, .048, .037, 0., 2*-.018, -.017, -.015,
     8  -.034, -.029, -.017, -.022, .037, 0., 2*-.015, -.018, -.022,
     9  -.027, -.03, -.024, 0., .037, 0., 2*-.016, -.017, 2*-.021, -.025
     X  , -.024, .006, .037, 0., 2*-.016, -.017, -.018, -.02, -.021,
     1  -.024, .006, .037, 0., 3*-.017, -.019, -.02, -.021, -.023, -.009
     2  , .037, 0., 3*-.017, -.019, -.02, 2*-.021, -.026, .037, 0.,
     3  3*-.017, -.019, -.021, -.022, -.023, -.031, -.002, 0., 3*-.017,
     4  -.019, -.021, -.023, -.027, -.036, .011, 0., 3*-.018, -.019,
     5  -.022, 2*-.027, -.043, .011, 0., 2*-.018, -.019, -.021, -.024,
     6  2*-.027, -.054, .011, 0., 2*-.018, -.019, -.021, -.024, -.028,
     7  -.031, -.063, .011, 0., 2*-.019, -.02, -.021, -.324, -.029,
     8  -.025, -.071, .011, 0. /
      DATA CM3B / 2*-.018, -.019, 2*-.02, -.025, -.037, -.081, .011, 0.,
     1  2*-.018, 2*-.019, -.017, -.02, -.039, -.089, .011, 0., 3*-.018,
     2  2*-.015, -.02, -.047, -.089, .011, 0., 2*-.017, -.016, -.011,
     3  -.01, -.022, -.061, -.089, .011, 0., 2*-.015, -.013, -.006, -.01,
     4  -.029, -.074, -.089, .011, 0., 2*-.014, -.009, -.006, -.012,
     5  -.04 , -.086, -.089, .011, 0., 2*-.097, -.026, -.022, -.04,
     6  -.06, -.1, -.089, .011, 0.,  2*-.099, -.055, -.053, -.074, -.08,
     7  -.116, -.089, .011, 0., 2*-.1, 2*-.083, -.098, -.101, -.116,
     8  -.089, .011, 0., 2*-.101, -.103, -.107, -.112, -.115, -.116,
     9  -.089, .011, 0., 2*-.1  , -.115, -.107, -.112, -.115, -.116,
     X  -.089, .011, 0., 2*-.118, -.123, -.107, -.112, -.115, -.116,
     1  -.089, .011, 0., 2*-.121, -.123, -.107, -.112, -.115, -.116,
     2  -.089, .011, 0., 2*.045 ,  .047,  .057, -.045, -.028, -.007,
     3  .048, .037, 0.  /
      DATA CL4  / 0. ,  .3  ,  .4  ,  .5  ,  .6  ,  .66 ,  .71 ,    .76
     1 ,  .91 ,  .85 , 1.   ,-180. ,  -6. ,  -5. ,  -4. ,  -3. ,  -2.
     2 ,  -1. ,  .0  , 1.   ,   2. ,   3. ,   4. ,   5. ,   6. ,   7.
     3 ,   8. , 9.   ,10.   ,  11. ,  12. ,  13. ,  14. ,  15. ,  16.
     4 , 180.
     5 ,11*. ,       2* -.541, -.554, -.578, -.612, -.701, -.76 ,   -.79
     6 , -.68 , -.575,  .0  ,       2* -.434, -.448, -.473, -.513,   -.57
     7 , -.63 , -.68 , -.56 , -.51 ,  .0  ,2*-.34, -.352, -.368,   -.401
     8 , -.429, -.492, -.53 , -.44 , -.435,  .0  ,2*-.25, -.255,   -.26
     9 , -.276, -.28 , -.29 , -.36 , -.39 , -.295,  .0  ,       2* -.116
     X , -.12 , -.129, -.145, -.154, -.13 , -.193, -.121, -.135,    .0
     1 , -.012, -.012, -.011, -.015,2*-.02,  .02 , -.016,  .001,    .025
```

ARAP06 FORTRAN Listings (Continued)

```
     2 .  .0  .  .071.  .091.  .095.  .101.  .111.  .118.  .17 .    .15
     3 .  .217.  .185.  .0  .2*.207.  .216.  .228.  .255.  .28 .    .32
     4 .  .335.  .36 .  .345.  .0  .2*.323.  .334.  .357.  .389.    .416
     5 .  .461.  .49 .  .51 .  .49 .  .0  .  .92 .  .42 .  .43 .  .48  /
      DATA CL4B /
     6    .52 .  .565.  .61 .  .625.  .65 .  .595.  .8  .  .595.    .545
     7 .  .569.  .634.  .665.  .712.  .767.  .774.  .765.  .66 .    .0
     8 .2*.66 .  .67 .  .7  .  .78 .  .84 .  .89 .  .875.  .84 .    .715
     9 .  .9  .2*.772.  .794.  .81 .  .916.  .97 .  .987.  .956.    .91
     X .  .74 .  .0  .2*.878.  .887.  .91 .  1.041. 1.075. 1.06 .    .99
     1 .  .945.2*.0  .2*.95 .  .937.  .981. 1.148. 1.15 . 1.126.   1.02
     2 .  .97 .2*.0  .2*.988.  .98 .  .99 . 1.21 . 1.199. 1.178.   1.055
     3 .3*.0  .  1.079. 1.009.  .994.  .995. 1.233. 1.142. 1.218. 4* .0
     4 .  1.216. 1.016.  .99 .  .97 .  1.12 .  .97 .  1.254.4*.0  . 2*1.004
     5 .  .986.  .96 .  .94 .  .9  .  1.2  .4*.0  .2*.98 .  .984.    .975
     6 .  .84 .  .865.  1.12 .4*.0  .2*.955.  .982.  .995.  .8  .    .835
     7 .  1.02 .4*.0  .2*.92 .9*.0  .2*.888.9*.0  .11*.0 /
      DATA CD4   / 0. .  .3  .  .4  .  .5  .  .6  .  .66 .  .71 .    .76
     1 .  .81 .  .85 .  1.   .-100. .  -6. .  -5. .  -4. .  -3. .  -2.
     2 .  -1. .  .0  .  1.  .   2. .   3. .   4. .   5. .   6. .   7.
     3 .   8. .  9.   .10.  .  11. .  12. .  13. .  14. .  15. .  16.
     4 .  100.
     5 .11*0. .2*.04 .  .047.  .06 .  .057.  .054.  .075.  .088.    .113
     6 .  .149.  .309.2*.025.  .028.  .036.  .033.  .336.  .05 .    .073
     7 .  .094.  .175.  .325.3*.009.  .016.  .019.  .07 .  .024.    .097
     8 .  .067.  .104.  .303.2*.007.  .008.  .011.  .01 .  .011.    .012
     9 .  .027.  .039.  .077.  .235.3*.007.3*.008.2*.009.  .015.    .049
     X .  .235.6*.037.2*.006.  .007.  .008.  .012.  .049.  .235. 3* .005
     1 .5*.006.  .021.  .049.  .235.7*.006.  .029.  .038.  .072.    .237
     2 .7*.006.  .015.  .056.  .095.  .246.2*.007.3*.006.2*.007.    .029
     3 .  .08 .  .121.  .25 .2*.008.  .007.2*.008.  .009.  .011.    .05
     4 .  .105.  .147.  .261.3*.008.  .011.  .015.  .017.  .025.    .072
     5 .  .128.  .167.  .271.2*.009.  .01 .  .013.  .019.  .028.  .049 /
      DATA CD4B /
     6    .098.  .149.  .186.  .204.2*.011.  .023.  .03 .  .04 .    .044
     7 .  .078.  .124.  .175.2*.0  .2*.015.  .052.  .056.  .061.    .074
     8 .  .115.  .155.  .203.2*.0  .2*.067.  .076.  .09 .  .098.    .104
     9 .  .143.  .18 .3*.0  .2*.107.  .117.  .123.  .108.  .141.    .17
     X .  .214.3*.0  .2*.147.  .152.  .154.  .153.  .17 .  .201. 4* .0
     1 .3*.181.  .184.  .191.  .208.  .243.4*.0  .2*.202.  .203.    .204
     2 .  .224.  .241.  .265.4*.0  .2*.227.  .22 .  .223.  .25 .    .267
     3 .  .286.4*.0  .2*.24 .9*.0  .2*.252.9*.0  .11*.0 /
      DATA CM4  / 0.. .301. .401. .501. .602. .652. .703..755..839. 1..
     1  -100.. -6.. -5.. -4.. -3.. -2.. -1.. 0.. 1.. 2.. 3.. 4.. 5..
     2  6.. 7.. 8.. 9.. 10.. 11.. 12.. 13.. 14..       16.. 20.. 100..
     3  2*-.021. -.018. -.016. -.022. -.041. -.029. .02. .1. .0.
     4  2*-.021. -.018. -.016. -.022. -.041. -.029. .02. .1. .0.
     5  2*-.024. -.025. -.027. -.03 . -.038. -.035. -.024. .1. 0..
     6  7*-.02 . -.021. -.024. -.029. -.034. -.038. -.027. .1. 0..
     7  3*-.02 . -.023. -.026. -.03 . -.033. -.032. .1. 0.. 2*-.019.
     8  -.02. -.022. -.024. -.026. -.028. -.034. -.018. 0.. 3*-.019.
     9  -.021. -.024. -.026. -.028. -.03. -.05. 0.. 3*-.02. -.022.
     X  -.024. -.026. -.028. -.032. -.085. 0.. 2*-.02. -.021. -.023.
     1  -.024. -.026. -.028. -.039. -.127. 0.. 2*-.07. -.021. -.023.
     2  -.025. -.027. -.029. -.051. -.159. 0.. 2*-.02. -.021. -.022.
```

ARAP06 FORTRAN Listings (Continued)

```
     3  -.025, -.026, -.029, -.060, -.159, 0., 2*-.019, -.021, -.022,
     4  -.024, -.025, -.03 , -.084, -.159, 0., 2*-.019, 2*-.02, -.019,
     5  -.02 , -.036, -.103, -.159, 0., 2*-.019, -.018, -.017, -.012,
     6  -.015, -.045, -.123, -.159, 0., 2*-.018, -.016, -.013, -.004,
     7  -.016, -.06 , -.123, -.159, 0., 2*-.014, -.012, -.013, -.001,
     8  -.022, -.077, -.123, -.159, 0.  /
      DATA CMOB/2*-.015, -.03 , -.034, -.003, -.03, -.089, -.123, -.159,
     1  0., 2*-.042, -.055, -.064, -.008, -.037, -.097, -.123, -.159,
     2  0., 2*-.075, -.082, -.091, -.046, -.037, -.095, -.123, -.159,
     3  0., 2*-.1, -.106, 2*-.11, -.037, -.085, -.123,-.159, 0., 2*-.110
     4  , -.109, -.113, -.11, -.037, -.085, -.123, -.159, 0., 2*-.126,
     5  -.11, -.113, -.11, -.037, -.085, -.123, -.159, 0., 2*-.121,
     7  -.11, -.113, -.11, -.037, -.085, -.123, -.159, 0., 2*-.121,
     8  -.11, -.113, -.11, -.037, -.085, -.123, -.159, 0., 2*-.021,
     9  -.018, -.016, -.022, -.041, -.079, .02, .1, 0.  /
      DATA  CALD / 1.,2.,3.,4.,5.,6.,7.,8.,9.,  0.,1.,2.,3.,4.,6.,8.,
     1  10.,12.,14.,16.,18.,20.,22.,24.,  .196, .393, .589, .785, .982,
     2  1.178, 1.374, 1.571, 1.767,  .196, .19, .56 , .738, .923,1.119,
     3  1.333, 1.539, 1.749,  .186,  .184, .53, .683, .864,1.06 ,1.278,
     4  1.492, 1.714,  .184,  .161,  .424,.528, .667, .872,1.113,1.366,
     5  1.676,  .155,  .126,  .43 ,  .377, .51, .73 , .962,1.225,1.484,
     6   .124,  .073,  .141,  .196,  .394,.507, .756,1.052,1.361, .004,
     7   .02 , -.061,  .0  ,  .118,  .318,.577, .88 ,1.202, .045,-.023,
     8  -.141, -.157, -.069,  .118,  .371,.691,1.042, .018,-.065,-.259,
     9  -.298, -.236, -.071,  .192,  .518,.883,-.016,-.114,-.418,-.495,
     X  -.461, -.283,  .0  ,  .314, .707,-.055,-.179,-.601,-.715,-.717,
     1  -.565, -.289,  .078,  .477,-.128,-.255,-.772,-.935,-.972,-.848,
     2  -.591, -.204,  .212, -.224, -.36, -1.001, -1.186, -1.247,
     3  -1.143,-.838, -.408,  .071,-.128,-.225,-.772,-.935,-.972,-.848,
     4  -.591, -.204,  .212, -.055,-.179,-.691,-.715,-.717,-.565,-.289,
     5   .078,  .477  /
      END
                          CARD COUNT       252
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE CLCDR (SMH,AL ,R,CL,CD,CM,    OK,CAD)
      COMMON /LLDD/ SM ,ALP
      DIMENSION ILT(3),Y(3),ST(3),      CURVE(220),DSX(30,3),NX(3)
      EQUIVALENCE           (Y(1),CL),(Y(2),CD),(Y(3),CM)
      DATA  NX /39,32,39/, ILT /0, 78,142/, ST/.9,.7,.45/, DSX/10.,
     8  2*5.,10.,2*5.,2*10.,20.,2*10.,20.,3*10.,2*5.,8.,8*2.,4.,2*2.,
     9  4.,3*2.,3.,5.,2*10.,130.,  20.,10.,2*5.,2*10.,20.,2*10.,20.,
     X  3*10.,2*5.,8.,3*2.,10.,4.,2*2.,4.,3*2.,3.,15.,10.,130.,7*0.,
     1  20.,3*10.,50.,6*10.,9.95,3*2.01,2.02,2.,3*2.02,1.99,2.,2.02,
     2  2.98,3.02,1.98,5.,4.95,3*10.,20.,2*10.,40.,10.,20.,10.  /
      DATA CURVE /-180.,-170.,-165.,-160.,-150.,-145.,-140.,-130.,-120.,
     1 -100.,-90.,-80.,-60.,-50.,-40.,-30.,-25.,-20.,-12.,-10.,-8.,-6.,
     2 -4.,-2.,0.,2.,4.,8.,10.,12.,16.,18.,20.,22.,25.,30.,40.,50.,180.,
     3 .003,.5,.49,.28,.13,.35,.52,.53,.38,.0,-.26,-.5,-.94,-1.,-.9,
     4 -.62,-.74,-.86,-.6,-.55,-.49,-.325,-.15,.038,.26,.42,.64,.935,
     5 1.01,1.105,1.085,1.175,1.011,1.024,1.04,.72,.97,1.,0.,    -180.,
     6 -160.,-150.,-145.,-140.,-130.,-120.,-100.,-90.,-80.,-60.,-50.,
     7 -40.,-30.,-25.,-20.,-12.,-10.,-9.,-6.,4.,9.,10.,12.,16.,18.,20.,
     8 22.,25.,40.,50.,180.,.03,.3,.45,.68,.95,1.2,1.51,2.,2.04,1.86,
     9 1.47,1.22,.86,.39,.27,.17,.038,.023,.017,2*.014,.016,.026,.042,
     X .1,.133,.174,.21,.26,.7,1.1,0.,  -180.,-160.,-150.,-140.,-130.,
     1 -80.,-70.,-60.,-50.,-40.,-30.,-20.,-10.05,-8.04,-5.03,-4.02,-2.,
     2 0.,2.02,4.04,6.06,8.05,10.05,12.07,15.05,18.07,20.05,25.05,30.,
     3 40.,50.,60.,80.,90.,100.,140.,150.,170.,180.,.2,.27,.34,.5,.56,
     4 .66,.63,.49,.31,.17,.05,-.01,-.0524,-.0587,-.0552,-.0525,-.0527,
     5 -.055,-.0596,-.0612,-.0611,-.0608,-.0647,-.0697,-.0706,-.067,
     6 -.0571,-.0641,-.1,-.165,-.255,-.395,-.72,-.79,-.78,-.695,-.645,
     7 -.41,-.37 /
C
      SM = AMAX1(OK,1.)
      ALPT = AMOD(AL ,180.)
      ALP = AMIN1(24.,ABS(ALPT))
      CALL TABINT(CAD,10)
      ALP = ALPT
      SM = SMH
      DO 10 K=1,3
      IF (R.LT.ST(K)) GO TO 10
      L = 10-3*K
      CALL TABINT (CL,L)
      CALL TABINT (CD,L+1)
      CALL TABINT (CM,L+2)
      RETURN
   10 CONTINUE
      IF (R.LT. .075) GO TO 25
      DO 15 N=1,3
      IL=ILT(N)
      NB=IL+2
      NL = NX(N) + IL
      NU=NL-1
      DO 1 K    =NB,NU
      IF    (ALP.LT.CURVE(K)) GO TO 3
    1 CONTINUE
      K    = NL
    3 KX= K  -IL-1
      M = K + NX(N)
```

```
      Y(N)= CURVE(M-1)+(ALP-CURVE(K-1))/DSX(KK,N)*(CURVE(M)-CURVE(M-1))
   15 CONTINUE
      RETURN
   25 DO 12 N=1,3
   12 Y(N) = 0.
      RETURN
      END
                          CARD COUNT        62
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE FLAPN1
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON / INPU / AM(10),DUM5(15),R(10),VB(10),FPC(29),AIC(15),
     1 EPF(10),C(17),  AMG(10),S,U,ZM,DUMA(55),
     2 GVM(10),DUM2(81),CNB(3),SNB(3)
      COMMON /AIRLD/ DUM7(10),ACCPM(3),ACSPP(3),VIN,CTMP,STMP,SSPM,CPSIP
     1 ,     DUMD(261),DHF(10,3),DO(10,3), DRADF(10,3), DTWST(10,3)
      COMMON /ELXOT/ FA1X,FA2X,FA3X,FA4X,FA5X,FA6X,PTTP
      COMMON /WINGT/ DUM9(5),X0,Y0,Z0
      COMMON/FLAP1/ CTM(10,3),STM(10,3),TM(10,3),SBET(3),CBET(3),BETD(3)
     1 ,BET(3),BETDD(3),THD(10,3),THDD(10,3),
     2 BCB(3),BDSQ(3),BSB(3),SCP,SSPP,DUM3(6),CSCP,CSSP,A1(3),A2(3),
     3 A3(3),CPSICP(3),SPSICP(3),B1(3),DUM1(15),ACSTP(3),CCTM(3)
      COMMON / FLAP2 /         SSB(10,3),SCB(10,3),CCB(10,3),CSB(10,3)
      COMMON / FLAP4 / CCPM, CSPP, PTP, TSP, CSTP, CCTM,PSID,PSIDD,SPSIP
     1 ,DUM8(5),    ATHDD, BTHDD,DUM4(10),AN(3)
      COMMON /FLAP6/ IT1,IDUM1(9),NS,NB,IDUM2(28),IT12
      DIMENSION F1(3)
      EQUIVALENCE     (A1(1),B11),(A1(2),B21),(B1(3),B31),(AN(1),AAN),
     4 (AN(2),BBN5),(AN(3),CCN6),(F1(1),F3),(F1(2),F5),(F1(3),F6),
     5 (A1(2),A12),(A1(3),A13),(A2(1),A21),(A2(2),A22),(A2(3),A23),
     6 (A3(1),A31),(A3(2),A32),(A3(3),A33),            (A1(1),A11)
C
      FA7X = X0*CTMP*CPSIP - Y0*SPSIP + Z0*STMP*CPSIP + ZM
      FA8X = X0*CSSP   + Y0*SPSICP(1)  + Z0*SSPP
      FA9X = X0*CSCP   + Y0*CPSICP(1)  + Z0*SSPM
      DO 20 J=1,3
      A1(J) = 0.D0
      A2(J) = 0.D0
      A3(J) = 0.D0
      B1(J) = 0.D0
      AN(J) = 0.D0
      F1(J) = 0.D0
   20 CONTINUE
      DO  90   J=1,NB
      SNBJ = SNB(J)
      CNBJ = CNB(J)
      V1    = (BETDD(J) - CCTM(J))*CBET(J) - (BETD(J) - ACCPM(J))*BSB(J)
      V6 = CBET(J)*PTTP - BSB(J)*PTP
      V2    = (CCTM - BETDD(J)*CNBJ)*SBET(J) - (BETD(J)*CNBJ - CCPM)*
     1 BCB(J) - V6*SNBJ
      V3    = V6*CNBJ - (BETDD(J)*SNBJ + CSTP)*SBET(J) - (BETD(J)*SNBJ
     1 + CSPP)*BCB(J)
      V4    =(CCPM - BETD(J)*CNBJ)*SBET(J) - CBET(J)*SNBJ*PTP
      V5    = CBET(J)*PTP*CNBJ - (BETD(J)*SNBJ + CSPP)*SBET(J)
      DO 90 I=1,NS
      RI = R(I)
      VBI = VB(I)
      AMI = AM(I)
      AICI = AIC(I)
      SSBM = CTM(I,J)*SNBJ   + SSB(I,J)*CNBJ
      CTMS = CTM(I,J)*CNBJ   - SSB(I,J)*SNBJ
      TBCCB      =-BETD(J)*  SSB(I,J) +  THD(I,J)*  CCB(I,J)
      ESRC       = RI   *CBET(J)
      TCT = THM(I,J)*CTM(I,J)
```

```
      TSQ = THD(I,J)**2
      FD    = THDD(I,J)  *  CSB(I,J) -   TSQ       *  SSB(I,J) * 2.*
     1   TCT       *BCB(J) - BDSQ(J)*  SSB(I,J)
      FAI9 = THD(I,J)*STH(I,J)  .
     1                   *CVBJ + (BCB(J)*STH(I,J) + CSB(I,J)*THD(I,J))*SNBJ
      FAI5 = (THD(I,J)*SNBJ - BCB(J)*CNBJ)*STH(I,J) - CSB(I,J)*THD(I,J)
     1  *CNBJ
      FAI1X = THDD(I,J)*CCB(I,J) - (TSQ+BDSQ(J))*SCB(I,J) -2.*TCT*BSB(J)
     1  - (STH(I,J)*ACSPP(J) - CSB(I,J)*ACCPM(J))*THD(I,J) + BCB(J)*
     2  STH(I,J)*ACCPM(J) + CTH(I,J)*ACSTP(J) + SSB(I,J)*CCTM(J)
      FAI2 = CTH(I,J)*ACSPP(J) + SSB(I,J)*(ACCPM(J) - BETD(J)) *THD(I,J)
     1  *CCB(I,J)
      F1A2 = (BETD(J)-ACCPM(J))*ESRC + YBI   *FAI2 + FA2X
      FAI3X = (THDD(I,J)*STH(I,J) + TSQ     *CTH(I,J))*SNBJ - FD*CNBJ +
     1  TACCR*CCPM + SCB(I,J)*CCTM + FAI3*PTP - CTHS*PTTP
      FAI4 = SCB(I,J)*CCPM - CTHS*PTP + FAI5
      F1A4 = V4*RI        + YBI  *FAI4 + FA4X
      FAI5X = (SSB(I,J)*BDSQ(J) - 2.*TCT     *BCB(J))*SNBJ - CSPP*TACCB
     1  - SCB(I,J)*CSTP - TSQ     *CTHS + PTP*FAI5 - PTTP*SSBB -
     2  (CSB(I,J)*SNBJ + STH(I,J)*CNBJ)*THDD(I,J)
      FAI6 =-SSBB*PTP - FAI3 - SCB(I,J)*CSPP
      F1A6 = V5*RI        + YBI  *FAI6 + FA6X
      FVD3 = V1*RI        + FAIX - F1A4*CCPM + F1A6*CSPP + (FAI1X -
     1  BETDD(J)*SSB(I,J))*YBI
      FVC5 = V2*RI        + FA3X - F1A6*PTP + F1A2*CCPM + (FAI3X -
     1  BETDD(J)*SCB(I,J)*CNBJ)*YBI
      FVD6 = V3*RI        + FA5X + F1A4*PTP - F1A2*CSPP + (FAI5X -
     1  BETDD(J)*SCB(I,J)*SNBJ)*YBI
      FAV3 = FAI1X - FAI4*CCPM + FAI6*CSPP
      FAV5 = FAI3X + FAI2*CCPM - FAI6*PTP + CTHS*PSIDD
      FAV6 = FAI5X + FAI4*PTP  - FAI2*CSPP+ SSBB*PSIDD
      AA    =          ESRC       - SSB(I,J)*YBI
      AB    = SBET(J)*RI          + SCB(I,J)*YBI
      R5F = AA    *CVBJ    - CTH(I,J)*SNBJ   *YBI
      R6F = AA    *SVBJ    + CTH(I,J)*CNBJ   *YBI
      R5E = ESRC     *CNBJ     - (SSB(I,J)*CVBJ     + CTH(I,J)*SNBJ   )*EPP(I)
      R6E = ESRC     *SNBJ     - (SSB(I,J)*SVBJ     - CTH(I,J)*CNBJ   )*EPP(I)
      F3 = F3 - FVD3    *AMI
      F5 = F5 - FVD5    *AMI
      F6 = F6 - FVD6    *AMI
      R3E     = SBET(J)*RI          + SCB(I,J)*EPP(I)
      AAN5 = DD(I,J)*SBET(J) + (R6F     *SSPP - R5F     *SSPM)*ANG(I) +
     1   (R6F     *SPSICP(I) - R5F     *CPSICP(I))*GYM(I)+AAN5 -
     2   (DHF(I,J)*CVBJ      + DRADF(I,J)*SNBJ   )*R5F - (DHF(I,J)*SNBJ     -
     3   DRADF(I,J)*CVBJ    )*R5E
      BBN5 = R6F     *DTHST(I,J) + (R6F     *SPSIP + AB     *CPSICP(I))*GYM(I)
     1   + (AB *SSPM - R6F     *SCP)*ANG(I)+BBN5 + DD(I,J)*CBET(J)*CNBJ
     2   + (DHF(I,J)*CVBJ      + DRADF(I,J)*SNBJ   )*R3E
      CCN6 =-R5F     *DTHST(I,J) - (AB     *SPSICP(I) + R5F     *SPSIP)*GYM(I)
     1   + (R5F *SCP - AB     *SSPP)*ANG(I)+CCN6 + (DHF(I,J)*SNBJ -
     2   DRADF(I,J)* CVBJ    )*R3E + DD(I,J)*CBET(J)*SNBJ
      FC5 = FVD5       + BETDD(J)*AB      *CNBJ     + R6F*PSIDD
      FD6 = FVD6       + BETDD(J)*AB      *SNBJ     - R5F*PSIDD
      FD3        = FVD3       - BETDD(J)*AA
      R3     = AB      + FA7X
```

ARAP06 FORTRAN Listings (Continued)

BELL HELICOPTER COMPANY

```
      R5    = R5F    + FA8X
      R6    = R6F    + FA9X
      A11 = (R5   *R5F    + R6   *R6F   )*AMI + (SSB(I,J)**2+CTH(I,J)**2
     1  )*AICI + A11
      SCBI = SCB(I,J)*AICI
      A21 = -R3  *R5F    *AMI + A21 + SSBB*SCBI
      A31 = -R3  *R6F    *AMI + A31 - CTHS*SCBI
      B11 = (R6*FD5 - R5*FD6)*AMI + (FAV5*CTHS + FAV6*SSBB)*AICI + B11
      B21 = (R3*FD6 - R6*FD3)*AMI + FAV6*SCBI - FAV3*CTHS*AICI + B21
      B31 = (R5*FD3 - R3*FD5)*AMI - FAV5*SCBI - FAV3*SSBB*AICI + B31
      A12 = CCB(I,J)*STH(I,J)*SNBJ  *AICI - (R5   *SNBJ   + R6   /2.D0)
     1  *SNBJ  *AB    *AMI + A12
      A13 = (R6   *CNBJ  **2 + R5   *SNBJ  /2.D0)*AB    *AMI + CTH(I,J)*
     1  CNBJ  *SCBI + A13
      IF (IT12.EQ.3) GO TO 90
      STSQ = STH(I,J)**2
      A22 = (R6   *AA    + R3   *SNBJ  *AB   )*SNBJ  *AMI + (SSB(I,J)*
     1  CTH(I,J)*.5D0 + STSQ     *SNBJ  )*SNBJ  *AICI + A22
      A23 = (R6   *AA    *CNBJ   - R3   *AB    *SNBJ  /2.D0)*AMI + A23 -
     1  (STH(I,J)*CSB(I,J)*CNBJ  **2 + STSQ      *SNBJ  /2.D0)*AICI
      A32 = (R3   *AB    /2.D0 - R5   *AA   )*SNBJ   *AMI + (STSQ/
     1  2.D0 - STH(I,J)*CSB(I,J)*SNBJ  )*SNBJ  *AICI + A32
      A33 =-(R5   *AA     + R3   *AB    *CNBJ  )*CNBJ  *AMI - (STSQ*
     1  CNBJ  **2 - STH(I,J)*CSB(I,J)*SNBJ  /2.D0)*AICI + A33
   90 CONTINUE
      IF (IT1.EQ.1) CALL RSTN
      B11= B11 + AAN3 + FA9X*F5 - FA8X*F6
      B21= B21 + BBN5 + FA7X*F6 - FA9X*F3
      B31= B31 + CCN6 + FA8X*F3 - FA7X*F5
      IF (IT12.EQ.3) RETURN
      A2233 = A22*A33 - A23*A32
      A1332 = A13*A32 - A12*A33
      A1223 = A12*A23 - A13*A22
      DETA = A11*A2233 + A21*A1332 + A31*A1223
      IF (IT12.EQ.2) GO TO 42
      PSIDD = ( A2233*B11 + A1332*B21 + A1223*B31) / DETA
      IF (IT12.EQ.1) RETURN
   42 BTHDD = (( A23*A31 - A21*A33)*B11 + (A11*A33 - A13*A31)*B21 + (
     1  A13*A21 - A11*A23)*B31) / DETA
      ATHDD = ((A21*A32 - A22*A31)*B11 + (A12*A31 - A11*A32)*B21 +
     1   (A11*A22 - A12*A21)*B31) / DETA
      RETURN
      END
                          CARD COUNT      153
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE FLAPN3
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 PLT(8)
      COMMON /INPU/ DUMJ(13),           THCH,DUML(81),ANG(10),
     1 S,W,ZM,RHO,DUMB(10),REALTR,DUM4(27),ALL,DUME(14),ANGY
      COMMON/FLAP1/ DUMA( 99),AFT(3),DUM3( 72),SCP,SSPP,SPSI,THPDC,STHPP
     1 ,THPHP,ALG,BLT,DUMI(3),A12,A13,A21,A22,A23,A31,A32,A33,DUMC(6),
     2 A11,A21,A31,ATHDDB(3),BTHDDB(3),PSIDDB(3)
      COMMON / FLAP4 / CCPM, CSPP, PTP, TSP, CSTP, CCTM,PSID,PSIDD,SPSIP
     1 ,ATH,BTH,PSI,ATHD,BTHD,          ATHDD, BTHDD,        DT,ATHDDE
     2 ,BTHDDE,PSIDDE,ATHDE,BTHDE,PSIDE,ATHE,BTHE,PSIE,AAN3,BBN5,CCN6,
     3 ,Y2,Y2, XPK(50,4),XTPK(50,4),DT6,T,TE,PKE(4),CCP
      COMMON [illegible]RLD/ DUM7( 25),CTHP,STHP,SSPM,CPSIP,
     4 CPS[illegible]D(29),                AVP(10,3),AVT(10,3),
     8 PH(10,3), AL(10,3),       A(10,3), DLIFT(10,3), DDRAG(10,3),
     3 DMOMT(10,3),DMF(10,3),MO(10,3),DRADF(10,3),DTMST(10,3)
     1 ,CLP(10,3),CDP(10,3),CMP(10,3),DT2,PH(10,3),PHE(10,3),DUM2(20),
     2 CPSIA(3),SPSIA(3)
      COMMON /WIN/                DUMM(234),SHPE(4),DUMG(8),
     1 TFFY,TFFX,TFFZ,TFMX,TFMY,TFMZ,EPW,ETW,THW
      COMMON /WINGT/ DUM6( 3),THPD,PSIPD,X0,Y0,Z0,THP,PSIP,DUMF(3),THPDD,
     1 PSIPDD,WVP(10),WVT(10),WPH(10),
     1 WAL(10),WCL(10),WCD(10),WL(10),WD(10),WT(10),WM(10),WD(10)
     3 ,XPDD( 9),YPDD( 9),ZPDD( 9),QMTRX(10),FMA(5)
      COMMON /FORCE/ AIM3,AIM5,AIM6,F3,F5,F6
      COMMON /FRIN/ DUMH(160),ZEDD(3,4),ZEDDE(3,4),ZEDE(3,4),ZEE(3,4),
     1 DUMM(172),ZE(3,4),ZED(3,4),DUMN(5),TBMT(4,4),TCMT(4,4)
      COMMON /FLEXOT/ ET(10,3),DUMR(60),ETW(10,3)
      COMMON /FLAP6/ IT1,IT2,IT3,IT4,IT5,IT6,IT7,IT8,IT9,IT10,NS,NR,
     1 NRSTN,NGOTO,NRV,JPSI,NRM,NB,NP,KPK(4),NM,NSW,NSHP,NPDD,IDUM1(2),
     2 NMT,IT11,KP,KSN(4),KSNE(4),IT12,NOAIR,IDUM2(6),NPT,IT14
      DIMENSION DATHDD(3),DBTHDD(3),DPSIDD(3),PK(4),EBMT(3,4),ECMT(3,4)
     1 ,TFK(5),DEG(5)
      EQUIVALENCE (TFFX,TFK(1)), (TFFZ,TFK(2)), (TFMX,TFK(3)),
     1 (TFMY,TFK(4)), (TFMZ,TFK(5))
C
      DATHDD(1) = ATHDD - ATHDDB(1)
      DBTHDD(1) = BTHDD - BTHDDB(1)
      DPSIDD(1) = PSIDD - PSIDDB(1)
      IF (DABS(DATHDD(1)     ).LT.DMAX1(.8D-2,.004*DABS(ATHDD)).AND.
     1    DABS(DBTHDD(1)     ).LT.DMAX1(.8D-2,.004*DABS(BTHDD)).AND.
     2    DABS(DPSIDD(1)     ).LT.DMAX1(.8D-2,.004*DABS(PSIDD))) GO TO 10
      CALL PSLOP (ATHDDB,DATHDD,JPSI)
      CALL PSLOP (BTHDDB,DBTHDD,JPSI)
      CALL PSLOP (PSIDDB,DPSIDD,JPSI)
      NGOTO =  0
      IF (JPSI.EQ.10) GO TO 10
      JPSI = JPSI+1
      RETURN
   10 NGOTO = -1
      IF (IT4.EQ.1) GO TO 95
      DO 94 J=1,NB
      DO 94 N=1,NBM
      IF (IT14.EQ.0) GO TO 20
      ZEDDE(J,N) =(ZEDD(J,N)  + ZEDDE(J,N))/2.D0
```

ARAP06 FORTRAN Listings (Continued)

```
      ZEDF (J,N) =(ZED  (J,N) + ZEDE(J,N))/2.D0
      ZEF  (J,N) =(ZE   (J,N) + ZEE(J,N))/2.D0
      GO TO 94
28    ZEDDF(J,N) = ZEDD(J,N)
      ZEDE (J,N) = ZED  (J,N)
      ZFE  (J,N) = ZE   (J,N)
94    CONTINUE
      IF (IT2.EQ.0) CALL TABINA(THCM,PSID)
      DO 30 J=1,NB
      DO 30 L=1,NMT
      BBMT(J,L) = 0.D0
      BCPT(J,L) = 0.D0
      DO 30 N=1,NBM
      BBMT(J,L) = BBMT(J,L) + ZEE(J,N)*TBMT(L,N)
      BCMT(J,L) = BCMT(J,L) + ZEE(J,N)*TCMT(L,N)
30    CONTINUE
95    IF (IT14.EQ.0) GO TO 40
      ATHDDE        =(ATHDDB(I) + ATHDDE)/2.D0
      BTHDDE        =(BTHDDB(I) + BTHDDE)/2.D0
      PSIDDE        =(PSIDDB(I) + PSIDDE)/2.D0
      ATHDE        =(ATHD   + ATHDE)/2.D0
      BTHDE        =(BTHD + BTHDE)/2.D0
      PSIDE        =(PSID + PSIDE)/2.D0
      ATHE        =(ATH + ATHE)/2.D0
      BTHE        =(BTH + BTHE)/2.D0
      PSIE =(PSI + PSIE)/2.D0
      GO TO 50
40    ATHDDF        = ATHDDB(I)
      BTHDDE        = BTHDDB(I)
      PSIDDE        = PSIDDB(I)
      ATHDE        = ATHD
      BTHDE        = BTHD
      PSIDE        = PSID
      ATHE        = ATH
      BTHE        = BTH
      PSIE = PSI
50    FFXA = SSPP*CPSI - SSPH*SPSI
      FFYA = SSPP*SPSI + SSPH*CPSI
      FFX = 0.D0
      FFY = 0.D0
      FFZ = 0.D0
      WBX=0.D0
      WBY=0.D0
      WBZ=0.D0
      DO 50 I=1,NS
      WBX  = WBX  - ANG(I)*FFXA
      WBY  = WBY  - ANG(I)*FFYA
      WBZ = WBZ - ANG(I)*SCP
      DO 50 J=1,NB
      FFX = FFX + DHF(I,J)*SPSIA(J) - DPADF(I,J)*CPSIA(J)
      FFY = FFY - DHF(I,J)*CPSIA(J) - DRADF(I,J)*SPSIA(J)
      FFZ = FFZ + DTHST(I,J)
      PHE(I,J) = PH(I,J)
50    CONTINUE
      CALL FORCES
```

ARAP06 FORTRAN Listings (Continued)

```
      WBX = WBX*3.
      WBY = WBY*3. - ANGY*CPSIP
      WBZ = WBZ*3. + ANGY*SPSIP
      TFFX = FFX + F5*CPSI - F6*SPSI + WBX
      TFFY = FFY + F5*SPSI + F6*CPSI + WBY
      TFFZ = FFZ + F3 + WBZ
      TFMX = (BBN5 + AIM5)*CPSI - (CCN6 + AIM6)*SPSI
      TFMY = (BBN5 + AIM5)*SPSI + (CCN6 + AIM6)*CPSI
      TFMZ = AAN3 + AIM3
      IF (IT12.EQ.0) GO TO 42
      IF (IT12.GT.1) TFMZ = TFMZ - B11 + A12*BTHDD + A13*ATHDD
      IF (IT12.EQ.2 .OR. IT4.EQ.0 .AND. IT12.NE.2) GO TO 42
      XM = B21 - A21*PSIDD
      YM = B31 - A31*PSIDD
      TFMX = TFMX - XM*CPSI + YM*SPSI
      TFMY = TFMY - XM*SPSI - YM*CPSI
42    IF (IT1.EQ.0) GO TO 90
      NRSTN=0
      CALL RSTN
90    IF (IT5.EQ.9) GO TO 65
      IF (IT8.NE.2) GO TO 74
      DO 76 K=1,5
76    TFK(K) = TFK(K) + FMA(K)
74    CALL WING
      THPDP = THPD*SPSIP*PSIPD
      TSP  = THPDD*SPSIP + THPDC*PSIPD
      CCP = CPSIP*CTHP*ZM
      X2 = CCP - ALL
      Y2 = ZM*SCP
      XMB= X0 + CCP - ZM
      YMB= Y0 - ZM*SPSIP
      ZMB =-Z0 - Y2
65    IF (PSIE.LT. 6.283200) GO TO 16
      PSIE = PSIE - 6.283200
      NRV = NRV+1
      IF (NRV.NE.3 .OR. NOAIR.EQ.1) GO TO 75
      WRITE (6,60) ((I,J,AVP(I,J),AVT(I,J),PHI(I,J),
     1  AL(I,J),CLP(I,J),CDP(I,J),CMP(I,J),A(I,J),DMF(I,J),DD(I,J),
     2  DRADF(I,J),DTWST(I,J),DLIFT(I,J),DDRAG(I,J),DMOMT(I,J),I=1,NS),
     3  J=1,NB)
60    FORMAT ( //// T55,      'ROTOR AIRLOADS DETAILS'        // T16,
     1  'STAT', T26,'BLADE', T37,'VP', T46,'VT', T56,'PHI', T67,'AL',
     2  T77,'CL', T87,'CD', T97 ,'CM', T106,'MACH' / T37,'DMF', T46,'DD'
     3  , T55,'DRADF', T65,'DTWST', T75,'DLIFT', T86,'DDRAG', T96 ,
     4  'DMOMT' / (/ 8X,2I10,F12.0,F8.0,F12.2,F10.2,4F10.3 / 30X,F10.0,
     5  F8.0,F12.2,F10.2,3F10.0))
      IF (IT5.EQ.0 .OR. RHO.EQ.0.D0 .OR. U.EQ.0.D0) GO TO 75
      WRITE (6,70) (I,WVP(I),WVT(I),WPH(I),WAL(I),WCL(I),
     1  WCD(I),WL(I),WD(I),WT(I),WM(I),WQ(I),I=1,NSW)
70    FORMAT ( //// T55,      'WING AIRLOADS DETAILS'        // T6,'STAT'
     1  , T14,'VP', T25,'VT', T36,'PH', T45,'AL', T54,'CL', T63,'CD',
     2  T74,'WL', T85,'WD', T96,'WT', T107,'WM', T118,'WQ' // (I8,F9.2,
     3  F11.0,2F10.2,2F9.3,F13.0,3F11.9,F12.0))
75    WRITE (6,10)
10    FORMAT ( ///////)
```

ARAP06 FORTRAN Listings (Continued)

```
      CALL HEADNG
16    DEG(1) = PSIDE/.10472
      DEG(2) = PSIE/.017453
      DEG(3) = BLG /.017453
      DEG(4) = BLT /.017453
      DEG(5) = AFT(1)/.017453
      WRITE (6,12) T,(DEG(K),K=1,5),FFX,FFY,FFZ,WBX,WBY,WBZ,F5,F6,F3,
     1 AIM5,AIM6,AI93,TFFX,TFFY,TFFZ,TFMX,TFMY,TFMZ
12    FORMAT (/5X,F13.3,F10.0,F10.1,3F10.3,6F10.1,(/5X,6F10.0,6F10.1))
      IF (IT4.EQ.1) GO TO 32
      WRITE (6,34) ((BBMT(J,L),BCMT(J,L),J=1,NB),L=1,NMT)
      WRITE (6,95) (ET(NS,J),ETN(NS,J),J=1,NB),((ZEE(J,N),J=1,NB),
     1 N=1,NBN)
32    IF (IT9.EQ.0) GO TO 62
      PK(1) = DEG(5)
      IF (IT5.EQ.0) GO TO 56
      PK(2) = -ETW
      PK(3) = EPW
      PK(4) = SHPE(1)
56    DO 60 K=1,KP
      IF (KPK(K).GT.40) GO TO 60
      KSN(K) = 1
      IF (PK(K).LT.PKE(K)) KSN(K) = -1
      IF (KSN(K)*KSVF(K).GE.0) GO TO 59
      KK = KPK(K) + 1
      KPK(K) = KK
      XPK(KK,K) = PKE(K)
      XTPK(KK,K) = TE
59    PKE(K) = PK(K)
      KSNF(K) = KSN(K)
60    CONTINUE
62    IF (IT5.EQ.0) GO TO 80
      PSIPN = -PSIP
      ETWN = -ETW
      WRITE (6,95) THP,PSIPN,XHB,YHB,ZHB,ETWN,EPW,THW,(S4PE(J),J=1,NSHP)
85    FORMAT (    5X,12F10.3)
      IF (IT11.EQ.1) WRITE (6,34) (CNTPX(I),I=1,10)
34    FORMAT (5X,12F10.0)
      IF (IT10.EQ.1) WRITE (6,85) (   XPDD(K),YPDD(K),ZPDD(K),K=1,NPDD)
80    IF (IT6.EQ.0) GO TO 82
      NP = NP+1
      IF (NP.LT.NPT) GO TO 82
      NP = 0
      NR = NR + 1
      PLT(1) = T
      GO TO (20,22,24), IT6
20    PLT(2) = DEG(1)
      PLT(3) = DEG(3)
      PLT(4) = DEG(4)
      PLT(5) = TFFX
      PLT(6) = TFFY
      PLT(7) = TFFZ
      PLT(8) = TFMZ
      GO TO 26
22    PLT(2) = ET(NS,1)
```


ARAP06 FORTRAN Listings (Continued)

```
      PLT(3) = ETN(NS,1)
      PLT(4) = BBMT(1,1)
      PLT(5) = BCMT(1,1)
      PLT(6) = BBMT(1,2)
      PLT(7) = BCMT(1,2)
      PLT(8) = DEG(5)
      GO TO 26
   24 PLT(2) = ETNM
      PLT(3) = EPW
      PLT(4) = THW
      PLT(5) = XPDD(1)
      PLT(6) = YPDD(1)
      PLT(7) = ZPDD(1)
      PLT(8) = XPDD(2)
   26 WRITE (1) (PLT(K),K=1,8)
   82 TE = T
      IF (T.LT.REALTM .AND.PSIDE.GT. -2.D0) GO TO 92
      NGOTO = 1
      IF (PSIDE.NE.0.D0 .OR. NOAIR.EQ.1) GO TO 92
      WRITE (6,53) (((I,J,AVP(I,J),AVT(I,J),PHI(I,J),
     1 AL(I,J),CLP(I,J),CDP(I,J),CMP(I,J),A(I,J),DHF(I,J),DQ(I,J),
     2 DRADF(I,J),DTHST(I,J),DLIFT(I,J),DDRAG(I,J),DMOMT(I,J),I=1,NS),
     3 J=1,NB)
      IF (ITS.EQ.0 .OR. RHO.EQ.0.D0 .OR. U.EQ.0.D0) GO TO 92
      WRITE (6,77) (I,WVP(I),WVT(I),WPM(I),WAL(I),WCL(I),
     1 WCD(I),WL(I),WD(I),WT(I),WH(I),WQ(I),I=1,NSW)
   92 RETURN
      END
                           CARD COUNT      248
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE FLEXBD
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /INPU/ AM(10),DUM5(5),THT(10),R(10),VB(10),DUM6(55),AMG(10)
     1 ,DUM8(58),                 GVM(10),DUM3(81),CNB(3),SNB(3)
      COMMON /AIPLD/ DUMC(15),ASSPM(3),ACCPM(3),ACSPM(3),DUM2(266),
     1  DMF(10,3),DD(10,3),DRADF(10,3),DTMST(10,3)
      COMMON / FRIN / BE(10,4),RI(10,4),VBT(10,4),THM(10,4),ZEDD(3,4),
     1  ZEDDE(3,4),ZEDE(3,4),            ZEE(3,4),
     2  BEM(4),   BOMSOM(   4),BOMZM2(   4),AMB(10,4),AMVBT(10,4),
     3  AMX(10,4),AIT(10,4),ZE(3,4),ZED(3,4)
      COMMON /FLEXOT/ ET(10,3),ETD(10,3),ETDD(10,3),ETN(10,3),ETND(10,3)
     1  ,ETNDD(10,3),ETV(10,3),ETVD(10,3),ETVDD(10,3),HRETSQ(10,3),CB,SB
      COMMON /FLXOT/ FA1X,FA2X,FA3X,FA4X,FA5X,FA6X
      COMMON /FLAP1/ DUME(174),SCP,DUMF(18),
     1  CPSICP(3),SPSICP(3),DUMI(12),THIN(3),ASSPP(3),ACSTP(3),CCTN(3)
      COMMON /FLAP2/ DUM4(120),PSIDA(3),THDD(3)
      COMMON /FLAP4/ CCPM,CSPP,PTP,TSP,CSTP,CCTM,PSID,PSIDD,SPSIP,
     1  DUMA(7),DT,DUMD(414),DT6
      COMMON /FLAP6/ IDUM1(10),NS,NB,IDUM2(4),NBM,IDUM3(6),NM,IDUM4(16),
     1  IT12,IDUM5(2),KOR(4)
      DIMENSION CTMB(10,3), ZEDDC(3,4),DZEDD(3,4),BD(3),    DTMSTB(10,3)
     1  ,DRADFB(10,3),DMFB(10,3),FO(3),ZDD(3,3,4),DZDD(3,3,4),
     2  F1A1B(10,3),F1A3B(10,3),F1A5B(10,3),FA11(10,3),STMB(10,3)
     4  ,S5(3),S6(3),S7(3),S8(3),S9(3),S10(3),S11(3),S12(3)
C
      DO 55 J=1,NB
      SNBJ = SNB(J)
      CNBJ = CNB(J)
      S5(J) = CB*CCPM + SB*PTP*SNBJ
      S6(J) = SB*CCPM - CB*PTP*SNBJ
      S7(J) = SB*CSPP - CB*PTP*CNBJ
      S8(J) = CB*CSPP + SB*PTP*CNBJ
      S9(J) = SB*CCTM + CB*TSP*SNBJ
      S10(J)= CB*CCTM - SB*TSP*SNBJ
      S11(J)= SB*CSTP + CB*TSP*CNBJ
      S12(J)= CB*CSTP - SB*TSP*CNBJ
      DO 55 I=1,NS
      DTMSTB(I,J) =  DTMST(I,J) - AMG(I)*SCP + GVM(I)*SPSIP
      DRADFB(I,J) =  DRADF(I,J) + AMG(I)*ASSPP(J) +     GVM(I)*SPSICP(J)
      DMFB(I,J) = DMF(I,J) - AMG(I)*ASSPM(J)           + GVM(I)*CPSICP(J)
   55 CONTINUE
      DO 60 K=1,10
      DO 61 J=1,NB
      DO 61 N=1,NBM
      ZED(J,N) = ZEDE(J,N) + (ZEDD(J,N)+ZEDDE(J,N))*DT
      ZE(J,N)  = ZEE(J,N)  + (ZED(J,N)+ZEDE(J,N))*DT + (ZEDD(J,N) -
     1  ZEDDE(J,N))*DT6
   61 CONTINUE
      DO 72 J=1,NB
      SNBJ = SNB(J)
      CNBJ = CNB(J)
      DO 72 I=1,NS
      ET   (I,J) = 0.D0
      ETD  (I,J) = 0.D0
      ETDD (I,J) = 0.D0
```

ARAP06 FORTRNA Listings (Continued)

```
      ETN  (I,J) = 0.D0
      ETND (I,J) = 0.D0
      ETNDD(I,J) = 0.D0
      ETV  (I,J) = 0.D0
      ETVD (I,J) = 0.D0
      ETVDD(I,J) = 0.D0
      DO 73 N=1,NMN
      ET   (I,J) = ET   (I,J) + BE(I,  N)*ZE  (J,N)
      ETD  (I,J) = ETD  (I,J) + BE(I,  N)*ZED (J,N)
      ETDD (I,J) = ETDD (I,J) + BE(I,  N)*ZEDD(J,N)
      ETN  (I,J) = ETN  (I,J) + XI(I,  N)*ZE  (J,N)
      ETND (I,J) = ETND (I,J) + XI(I,  N)*ZED (J,N)
      ETNDD(I,J) = ETNDD(I,J) + XI(I,  N)*ZEDD(J,N)
      ETV  (I,J) = ETV  (I,J) +THM(I,  N)*ZE  (J,N)
      ETVD (I,J) = ETVD (I,J) +THM(I,  N)*ZED (J,N)
      ETVDD(I,J) = ETVDD(I,J) +THM(I,  N)*ZEDD(J,N)
   73 CONTINUE
      THBD     = PSIDA(J) + ETVD(I,J)
      THB = THIN(J) + THT(I) + ETV(I,J)
      STHB(I,J) = DSIN(THB)
      CTHB(I,J) = DCOS(THB)
      TSQ     = THBD     **2
      RETETD       = (ET(I,J)*ETD(I,J) + ETN(I,J)*ETND(I,J)) / R(I)
      HRETSQ(I,J) = (ET(I,J)**2 + ETN(I,J)**2) / 2. / R(I)
      RETSQN       = (ETD(I,J)**2 + ETND(I,J)**2+ ET(I,J)*ETDD(I,J) +
     1 ETN(I,J)*ETNDD(I,J)) / R(I)
      TBCCB = THBD*CTHB(I,J)*CB
      ESRC        = ET(I,J)*SB      - R(I)*CB
      ETDSB       = ETD(I,J)*SB
      VBI = VB(I)
      S1 = CTHB(I,J)*SB*SNBJ + STHB(I,J)*CNBJ
      S3 = CTHB(I,J)*SB*CNBJ - STHB(I,J)*SNBJ
      S2 = STHB(I,J)*SB*SNBJ - CTHB(I,J)*CNBJ
      S4 = STHB(I,J)*SB*CNBJ + CTHB(I,J)*SNBJ
      FA12 = TBCCB - CSPP*S2 + CCPM*S4
      FA14 = STHB(I,J)*CCPM*CB + PTP*S2 - THBD*S3
      FA16 =-STHB(I,J)*CSPP*CB - PTP*S4 - THBD*S1
      FA11X = ETVDD(I,J)*CTHB(I,J) + (CCPM*S3 - CSPP*S1)*THBD -  CSTP*
     1 S2 + CCTM*S4   + (THDD(J)*CTHB(I,J) - STHB(I,J)*TSQ)*CB
      FA13X = TSQ*S4 - THDD(J)*S3 + TBCCB*CCPM + STHB(I,J)*CCTM*CB +
     1 PTP*THBD*S1 - TSP*S2
      FA15X = TSQ*S2 - THDD(J)*S1 - PTP*THBD*S3 + TSP*S4 - TBCCB*CSPP
     1 - STHB(I,J)*CSTP*CB
      F1A2 = (ESRC + HRETSQ(I,J)*CB)*ACCPM(J) + FA2X        - ETN(I,J)
     1 *ACSPP(J) - RETETD*SB + VBI*FA12 + ETD(I,J)*CB
      F1A4 = (R(I) - HRETSQ(I,J))*S6(J) + ET(I,J)*S5(J) + ETND(I,J)*
     1 SNBJ + VBI*FA14 + FA4X + (ETN(I,J)*PTP - ETDSB - RETETD*CB)*CNBJ
      F1A6 = (HRETSQ(I,J) - R(I))*S7(J) - ET(I,J)*S8(J) - ETND(I,J)*
     1 CNBJ + VBI*FA16 + FA6X + (ETN(I,J)*PTP - ETDSB - RETETD*CB)*SNBJ
      F1A1B(I,J) = (RETETD*CB +ETDSB)*ACCPM(J) + (HRETSQ(I,J)*CB +ESRC)*
     1 CCTM(J) - ETVD(I,J)*ACSPP(J) - ETN(I,J)*ACSTP(J) + VBI*FA11X +
     2 F1A6*CSPP - F1A4*CCPM + FA1X - RETSQN*SB
      F1A3B(I,J) = F1A2*CCPM-F1A6*PTP + (ETND(I,J)*PTP - ETN(I,J)*TSP-CB
     1 *RETSQN)*CNBJ + VBI*FA13X + FA3X - S6(J)*RETETD + (R(I) -
     2 HRETSQ(I,J))*S9(J) + ET(I,J)*S10(J) + ETD(I,J)*S5(J)
```

ARAP06 FORTRAN Listings (Continued)

```
      F1A5B(I,J) = (FTNDE(I,J)*PTP - ETN(I,J)*TSP - CB*RETSON)*SNBJ +VBI*
     1  FA15X + FA5X + ST(J)*RETETD - (R(I) - HRETSO(I,J))*S11(J) + F1A6
     2  *PTP - F1A7*CSPP - ET(I,J)*S12(J) - ETD(I,J)*SB(J)
      FA11(I,J) = (FA11X + FA16*CSPP - FA14*CCPM)*CTHB(I,J)*CB - (FA13X
     1  - FA16*PTP + FA12*CCPM)*S3 - (FA15X + FA14*PTP - FA12*CSPP)*S1
   72 CONTINUE
      DO 75 N=1,NBM
      DO 10 J=1,NB
      BC(J) = BOMZM2(N)*ZED(J,N) + BOMSOM(N)*ZE(J,N)
      FO (J  ) = 0.D0
      DO  20 I=1,NS
      BYBTCT          = BF(I,  N) + YBT(I,  N)*CTHB(I,J)
      BEXEN          = (BE(I,N)*ET(I,J) + XI(I,N)*ETN(I,J))/R(I)
      AMBX  = AMB(I,N) + AMVBT(I,N)*CTHB(I,J)
      FE    =                            AMBX*SB + AN(I)*BEXEN    *CB
      AX    = AMX(I,N) + AMVBT(I,N)*STHB(I,J)
      FO(J) = FO(J) + (BYBTCT*CB - SB*BEXEN)*DTHSTB(I,J) + (BYBTCT*SB
     1  + CB*BEXEN(N)*DRADFR(I,J) + DO(I,J)*THM(I,N) + (XI(I,N) + VBT(I,N)
     2  *STHB(I,J))*DHFB(I,J)                       - (SNB(J)*AX - CNB(J)*FE)
     3  *F1A3B(I,J) + (CNB(J)*AX + SNB(J)*FE)*F1A5B(I,J) - AIT(I,N)*
     4  FA11(I,J) - (AMBX*CB - AN(I)*BEXEN*SB)*F1A1B(I,J)
   20 CONTINUE
   10 CONTINUE
      IF (IT12.NE.3 .OR. PSIO.NE.0.D0) GO TO 40
      DO 46 J=1,NB
      ZEDDC(J,N) = FO(J)/BEM(N) - BD(J)
   46 CONTINUE
      GO TO 75
   40 IF (KDR(N).EQ.1) GO TO 32
      ZEDDC(1,N) = (FO(1) + FO(2) + FO(3))/3.D0/BEM(N) - BD(1)
      ZEDDC(2,N) = ZEDDC(1,N)
      ZEDDC(3,N) = ZEDDC(1,N)
      GO TO 75
   32 ZEDDC(1,N) = (2.D0*FO(1) - FO(2) - FO(3))/3.D0/BEM(N) - BD(1)
      OBCD = ((FO(2) - FO(3))/BEM(N) - BD(1))/2.D0 - BD(2)
      ZEDDC(2,N) = -.5*ZEDDC(1,N) + OBDD
      ZEDDC(3,N) = -.5*ZEDDC(1,N) - OBDD
   75 CONTINUE
      JN = 0
      DO 84 J=1,NB
      DO 84 N=1,NBM
      DZEDD(J,N) = ZEDDC(J,N) - ZEDD(J,N)
      IF(DABS(DZEDD(J,N)) .GT. DMAX1(5.D-4,.0005*DABS(ZEDDC(J,N))))
     1  JN = JN + 1
   84 CONTINUE
      IF (JN.EQ.0) GO TO 25
      DO 60 J=1,NB
      DO 86 N=1,NBM
      ZDD(1,J,N) = ZEDD(J,N)
      DZDD(1,J,N)= DZEDD(J,N)
      CALL PSLOP (ZDD(1,J,N),DZDD(1,J,N),K)
      ZEDD(J,N) = ZDD(1,J,N)
   86 CONTINUE
   60 CONTINUE
   25 RETURN
```

PAGE 4

```
      END                   CARD COUNT      166
```

ARAP06 FORTRAN Listings (Continued)

BELL HELICOPTER COMPANY

```
      SUBROUTINE FLEX1
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /FLAP6/ IT1,IT2,IT3,IT4,IDUM1(6),NS,NB
      COMMON /WINGT/ XOD,YOD,ZOD ,THPD,PSIPD,DUMB(5),XODD,YODD,ZODD
     1  ,THPDD,PSIPDD
      COMMON /FLXOT/ FA1X,FA2X,FA3X,FA4X,FA5X,FA6X,PTTP
      COMMON / INPU / AM(10),A0,A1,B1,THCH,TDEL3,THT(10),R(10),DUM9(72),
     1  ZH,DUM3(51),BTHA,DUM5(53),CTHT(10),STHT(10),DUM4(21),CNB(3),
     2  SNB(3)
      COMMON /FLEXOT/ ET(10,3),DUME(150),ETV(10,3),ETVD(10,3)
      COMMON /TAIR/              TDCT(10,3),STBSB(10,3),STBCB(10,3),
     1  CTHB(10,3),STHB(10,3),THRD(10,3),ESRC(10,3),ECPS(10,3)
      COMMON /AIRLD/ ACCP(3),ASCS(3),ACCS(3),ACSC(3),ASSP(3),
     1          ASSPM(3),ACCPM(3),ACSPP(3),VINDCA,      CTHP,STHP,SSPM,
     2  CPSIP,CPSI,DUM6(551),CPSIA(3),SPSIA(3)
      COMMON/FLAP1/ CTH(10,3),STH(10,3),TH(10,3),SBET(3),CBET(3),BETD(3)
     1  ,BET(3),BETDD(3),THD(10,3),THDD(10,3),
     2  BCB(3), BDSQ(3), BSB(3), SCP,SSPP,SPSI,THPDC,STHPP,THPDP,BLG,
     3  BLT,CSCP,CSSP,DUMD(9),CPSICP(3),SPSICP(3),DUMA(12),
     4                  THIN(3),ASSPP(3),ACSTP(3),CCTM(3),AIM,BIM
      COMMON / FLAP2 /                          SSB(10,3),SCB(10,3),
     1  CCB(10,3),CSB(10,3),              PSIDA(3),PSIDDA(3)
      COMMON /FLAP4/ CCPM,CSPP,PTP,TSP,CSTP,CCTM,PSID,PSIDD,SPSIP
     1  ,ATH,BTH,PSI,ATHD,BTHD,           ATHDD, BTHDD
C
      CSTPA = THPDD*CPSIP- THPDP - PSIPD*PSID
      CSTPB = THPDC*PSID + PSIPDD
      DO 70 J=1,NB
      CNBJ = CNB(J)
      SNBJ = SNB(J)
      THIM = ATH*CNBJ + BTH*SNBJ
      BET(J) = A0 + THIM
      SPSIA(J) = SPSI*CNBJ + CPSI*SNBJ
      CPSIA(J) = CPSI*CNBJ - SPSI*SNBJ
      BETD (J) = ATHD *CNBJ + BTHD *SNBJ
      BETDD(J) = ATHDD*CNBJ + BTHDD*SNBJ
      SBET(J) =DSIN(BET(J))
      CBET(J) =DCOS(BET(J))
      BCB(J) = BETD(J)*CBET(J)
      BSB(J) = BETD(J)*SBET(J)
      BDSQ(J) = BETD(J)**2
      A1PB1 = A1M*SPSIA(J) - B1M*CPSIA(J)
      A1PR1 = A1M*CPSIA(J) + B1M*SPSIA(J)
      PSIDA(J) = PSID*A1PB1
      PSIDDA(J) = PSIDD*A1PR1 + A1PB1*PSID*PSID
      BTT = BETDD(J)*TDEL3
      THIN(J) = THCH - A1PB1
      SPSICP(J) = SPSIA(J)*CPSIP
      CPSICP(J) = CPSIA(J)*CPSIP
      ACCP(J) = CPSIP*CBET(J) - SPSIP*SPSIA(J)*SBET(J)
      ASCS(J) = STHP*CPSIA(J)*SBET(J)
      ACCS(J) = CTHP*CPSIA(J)*SBET(J)
      ACSC(J) = CTHP*SPSIP*CPSIA(J) + STHP*SPSIA(J)
      ASSP(J) = SPSIP*SBET(J) - CBET(J)*SPSICP(J)
      ASSPM(J)= CTHP*SPSIA(J) - STHPP    *CPSIA(J)
```

ARAP06 FORTRAN Listings (Continued)

```
      ACCPM(J) = THPDC  *CPSIA(J) - PSIPD*SPSIA(J)
      ACSPP(J) = THPDC  *SPSIA(J) + PSIPD*CPSIA(J)
      ASSPP(J) = SPSIA(J)*STHPP + CPSIA(J)*CTHP
      ACSTP(J) = SPSIA(J)*CSTPA + CPSIA(J)*CSTPB
      CCTN (J) = CPSIA(J)*CSTPA - SPSIA(J)*CSTPB
      THINM = THIN(J) - THIM*TDEL3
      STHIM    = DSIN(THINM)
      CTHIM    = DCOS(THINM)
      DO 70 I=1,NS
      THDD(I,J) = PSIDDA(J) - BTT
      THD(I,J) = PSIDA(J)  - BETD(J)*TDEL3 + PIMA
      TH(I,J) = THT(I)+THINM
      STH  (I,J) = STHT(I)*CTHIM    + CTHT(I)*STHIM
      CTH  (I,J) = CTHT(I)*CTHIM    - STHT(I)*STHIM
      SSB(I,J)= STH(I,J)*SBET(J)
      SCB(I,J)= STH(I,J)*CBET(J)
      CCB(I,J)= CTH(I,J)*CBET(J)
      CSB(I,J)= CTH(I,J)*SBET(J)
      THPD(I,J) = THD(I,J) + ETVD(I,J)
      STHB(I,J) = STH(I,J)
      CTHB(I,J) = CTH(I,J)
      IF (IT4.EQ.1) GO TO 74
      THB      = TH(I,J) + ETY(I,J)
      STHB(I,J) = DSIN(THB)
      CTHB(I,J) = DCOS(THB)
   74 STBSB(I,J) = STHB(I,J)*SBET(J)
      STBCB(I,J) = STHB(I,J)*CBET(J)
      TDCT(I,J) = THBD(I,J)*CTHB(I,J)
      ESPC(I,J)  = ET(I,J)*SBET(J) - R(I)*CBET(J)
      ECPS(I,J)  = ET(I,J)*CBET(J) + R(I)*SBET(J)
   70 CONTINUE
      CSPP = ACSPP(1)
      SSPM = -ASSPM(1)
      CCPM = ACCPM(1)
      SSPP = ASSPP(1)
      PTP  = PSID - THPD*SPSIP
      CSTP = ACSTP(1)
      CCTM = CCTN(1)
      CSSP  = CTHP*SPSIP*SPSI - STHP*CPSI
      CSCP  = CTHP*SPSIP*CPSI + STHP*SPSI
      PTTP  = PSIDD - TSP
      FA3X = XODD*CSSP + (SPSICP(1)*CTHP*PSIPD + PSID*CSCP - THPD*SSPP)
     1  *XOD + SPSICP(1)*YODD + (CPSICP(1)*PSID - PSIPD*SPSIP*SPSI)*YOD
     2  + ZODD*SSPP + (THPD*CSSP + SPSICP(1)*PSIPD*STHP + PSID*SSPM)*
     3  ZOD + ZH*CCTM
      FA4X  = XOD*CSSP + YOD*SPSICP(1) + ZOD*SSPP + ZH*CCPM
      FA5X = XODD*CSCP + (CPSICP(1)*PSIPD*CTHP - PSID*CSSP - THPD*SSPM)
     1  *XOD + CPSICP(1)*YODD - (PSIPD*SPSIP*CPSI + PSID*SPSICP(1))*YOD
     2  + ZODD*SSPM + (THPD*CSCP + CPSICP(1)*PSIPD*STHP - PSID*SSPP)*
     3  ZOD - ZH*CSTP
      FA6X  = XOD*CSCP + YOD*CPSICP(1) + ZOD*SSPM - ZH*CSPP
      RETURN
      END
                          CARD COUNT       100
```

ARAP06 FORTARN Listings (Continued)

PAGE 1

```
      SUBROUTINE FORCES
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /FORCE/ AIM3,AIM5,AIM6,F3,F5,F6
      COMMON /INPU /  AM(10),DUM(15),R(13),VB(10),DUM5(23),AIC(13)
     2  ,DUM3(179),CNB(3),SNB(3)
      COMMON /FLXOT/ FA1X,FA2X,FA3X,FA4X,FA5X,FA6X,PTTP
      COMMON /FLAP1/ DUM2(90),                              SBET(3),CBET(3),
     1  BETD(3),BET(3),BETDD(3),THD(10,3),THDD(10,3),BCB(3),
     2  BDSQ(3),BSB(3),DUM1(43),ACSTP(3),CCTN(3)
      COMMON /FLAP4/ CCPM,CSPP,PTP,TSP,CSTP,CCTM
      COMMON /FLAP6/IDUM4(10),NS,NB
      COMMON / FLEXOT/ET(10,3),ETD(10,3),ETDD(10,3),ETN(10,3),ETND(10,3)
     1 ,ETNDD(10,3),ETV(10,3),ETVD(10,3),ETVDD(10,3),HRETSQ(10,3)
      COMMON /TAIR/               TDCT(10,3),STBSB(10,3),STBCB(10,3),
     1  CTHB(10,3),STHB(10,3),THBD(10,3),ESPC(10,3),ECRS(10,3)
      COMMON /AIRLD/ DUM6(21),ACSPP(3)
C
      AIM3 = 0.D0
      AIM5 = 0.D0
      AIM6 = 0.D0
      F3 = 0.D0
      F5 = 0.D0
      F6 = 0.D0
      DO 72 J=1,NB
      SNBJ = SNB(J)
      CNBJ = CNB(J)
      SJ = SBET(J)
      CJ = CBET(J)
      V1    = BCB(J)*CSPP + SJ     *CSTP + (BSB(J)*PTP - CJ     *PTTP)*
     1  CNBJ + (BETDD(J)*SBET(J) + BETD(J)*BCB(J))*SNBJ
      V2    = (BETD(J)*SNBJ*2.D0 + CSPP)*SJ      - CJ      *PTP*CNBJ
      V3    = (BETD(J)*SNBJ + CSPP)*BSB(J) - (BETDD(J)*SNBJ + CSTP)
     1  *CBET(J) - (BCB(J)*PTP + SBET(J)*PTTP)*CNBJ
      V4    = SJ     *PTP*CNBJ + CJ     *CSPP + BCB(J)*SNBJ*2.
      V5    = BETD(J) + CSPP*SNBJ - CCPM*CNBJ
      V6    = SJ     *CCPM - BSB(J)*CNBJ - CJ     *PTP*SNBJ
      V7    = CJ     *CCPM - BCB(J)*CNBJ + SJ     *PTP*SNBJ
      V8    = (BETD(J)*SNBJ + CSPP)*SJ      - CBET(J)*PTP*CNBJ
      V9    =-(BETD(J)*SNBJ + CSPP)*CJ      - SJ     *PTP*CNBJ
      V10 = BETDD(J) - CCTN(J)
      V11 = BSB(J)*2.D0*CNBJ - SJ*CCPM + CJ*PTP*SNBJ
      V15 = CCPM - BETD(J)*CNBJ
      V12 = SJ*CCTM - BETDD(J)*SJ*CNBJ + (BSB(J)*PTP - CJ*PTTP)*SNBJ +
     1  BCB(J)*V15
      V13 = (BCB(J)*PTP + SJ*PTTP)*SNBJ - BSB(J)*V15 + (CCTM - BETDD(J)
     1  *CNBJ)*CJ
      V14 = CJ*V15 + SJ*PTP*SNBJ - BCB(J)*CNBJ
      V15 = BCB(J) + V5*CJ
      DO 72 I=1,NS
      VBI = VB(I)
      AMI = AM(I)
      CTBSB = CTHB(I,J)*SBET(J)
      CTBCB = CTHB(I,J)*CBET(J)
      S1 = CTBSB     *SNBJ + STHB(I,J)*CNBJ
      S2 = STBSB(I,J)*SNBJ - CTHB(I,J)*CNBJ
```

ARAP06 FORTRAN Listings (Continued)

```
      S3 = CTBSB     *CNBJ - STHB(I,J)*SNBJ
      S4 = STBSB(I,J)*CNBJ + CTHB(I,J)*SNBJ
      TBDSQ    = THBD(I,J)**2
      RETETD       = (ET(I,J)*ETD(I,J) + ETN(I,J)*ETND(I,J)) / R(I)
      RETSQN       = (ETD(I,J)**2 + ETND(I,J)**2+ ET(I,J)*ETDD(I,J) +
     1  ETN(I,J)*ETNDD(I,J)) / R(I)
      TBCCB       =-BETD(J)*STBSB(I,J) + THBD(I,J)*CTBCB
      ETDSB       = ETD(I,J)*SJ
      FA11X   =THDD(I,J) *CTBCB       - TBDSQ      *STBCB(I,J) - 2.*
     1  TDCT(I,J)*BSB(J) - BETDD(J)*STBSB(I,J) - BDSQ(J)*STBCB(I,J)
     2  + ETYDD(I,J)*CTBCB        - (THBD(I,J)*S1 + BCB(J)*STHB(I,J)*
     3  SNBJ)*CSPP - CSTP*S2 + (THBD(I,J)*S3 + BCB(J)*STHB(I,J)*CNBJ)*
     4  CCPM + CCTM*S4
      FA13X =  -(THDD(I,J)*CTBSB        - TBDSQ      *STBSB(I,J) + 2.*
     1  TDCT(I,J)*BCB(J) - BDSQ(J)*STBSB(I,J) +BETDD(J)*STBCB(I,J))*CNBJ
     2 +(THDD(I,J)*STHB(I,J) + TBDSQ*CTHB(I,J))*SNBJ + CCPM*TBCCB +
     3  STBCB(I,J)*CCTM + (THBD(I,J)*S1 + BCB(J)*STHB(I,J)*SNBJ)*PTP +
     4  PTTP*S2 - ETYDD(I,J)*S3
      FA15X = TBDSQ      *S2 + BDSQ(J)*STBSB(I,J) - THDD(I,J)*S1 - TBCCB
     1  *CSPP - STBCB(I,J)*CSTP - (TDCT(I,J)*BCB(J)*2.D0 + BETDD(J)*
     2  STBCB(I,J))*SNBJ - (THBD(I,J)*S3 + BCB(J)*STHB(I,J)*CNBJ)*PTP
     3  - PTTP*S4 - ETYDD(I,J)*S1
      FA12 = TBCCB - S2*CSPP + S4*CCPM
      FA14 = S2*PTP + STBCB(I,J)*(CCPM - BETD(J)*CNBJ) - S3*THBD(I,J)
      FA16 = -S4*PTP - STBCB(I,J)*(CSPP + BETD(J)*SNBJ) - S1*THBD(I,J)
      F1A2      = FA12     *VBI + FA2X + ETD(I,J)*CJ - RETETD*SJ -
     1  ACSPP(J)                 *ETN(I,J) - (ESRC(I,J)+HRETSQ(I,J)*CJ)*V5
      F1A4      = FA14     *VBI + FA4X + ETND(I,J)*SNBJ + ET(I,J)*V7     +
     1  (R(I) - HRETSQ(I,J))*V6      + (ETN(I,J)*PTP - ETDSB -
     2  RETETD*CJ)*CNBJ
      F1A6      = FA16     *VBI + FA6X - ETND(I,J)*CNBJ + ET(I,J)*V9     +
     1  (HRETSQ(I,J) - R(I))*V8      + (ETN(I,J)*PTP - ETDSB -
     2  RETETD*CJ)*SNBJ
      FA3    = -ETD(I,J)*BSB(J) - SBET(J)*RETSQN - V15*RETETD + VBI*FA11X
     1  + FA1X + ETDD(I,J)*CBET(J) + (HRETSQ(I,J)*BSB(J) - BETD(J)*
     2  ECRS(I,J) - ETDSB)*V5 - (HRETSQ(I,J)*CJ + ESRC(I,J))*V10 +
     3  F1A6*CSPP - F1A4*CCPM - ACSPP(J)*ETND(I,J) - ACSTP(J)*ETN(I,J)
      FA5 = F1A2*CCPM - F1A6*PTP + ETNDD(I,J)*SNBJ + VBI*FA13X + FA3X
     1  + (ETND(I,J)*PTP + ETN(I,J)*PTTP - ETDD(I,J)*SJ - RETSQN*CJ)*
     2  CNBJ + RETETD*V11 + (R(I) - HRETSQ(I,J))*V12 + ET(I,J)*V13 +
     3  ETD(I,J)*V14
      FA6 = F1A4     *PTP - F1A2     *CSPP - ETNDD(I,J)*CNBJ + FA15X    *VBI
     1  + FA5X + (HRETSQ(I,J) - R(I))*V1      + RETETD    *V2     + ET(I,J)
     2  *V3      - ETD(I,J)*V4      + (ETND(I,J)*PTP - RETSQN    *CJ +
     3  ETN(I,J)*PTTP - ETDD(I,J)*SBET(J))*SNBJ
      AV3 = FA11X      + FA16     *CSPP - FA14     *CCPM
      AV5 = FA13X      - FA16     *PTP + FA12     *CCPM
      AV6 = FA15X      + FA14     *PTP - FA12     *CSPP
      R3 = ECRS(I,J) + STBCB(I,J)*VBI
      R5 = ETN(I,J)*SNBJ - ESRC(I,J)*CNBJ - S4*VBI
      R6 =-ETN(I,J)*CNBJ - ESRC(I,J)*SNBJ - S2*VBI
      AIM3 = (FA5*R6 - FA6*R5)*AMI + (S4*AV6 - S2*AV5)*AIC(I) + AIM3
      AIM5 = (FA6*R3 - FA3*R6)*AMI + (S2*AV3 + STBCB(I,J)*AV6)*AIC(I) +
     1  AIM5
      AIM6 = (FA3*R5 - FA5*R3)*AMI - (STBCB(I,J)*AV5 + S4*AV3)*AIC(I) +
```

ARAP06 FORTRAN Listings (Continued)

```
    1   AIM6
        F3 =-FA3   *AMI + F3
        F5 =-FA5   *AMI * F5
        F6 =-FA6   *AMI + F6
   72   CONTINUE
        RETURN
        END
                              CARD COUNT      117
```

ARAP06 FORTRAN Listings (Continued)

PAGE 1

```
      SUBROUTINE HEADNG
      COMMON /FLAP6/ IT1,IT2,IT3,IT4,IT5,IT6,IT7,IT8,IT9,IT10,IDUM(6),
     1  NBM,IDUM1(9),NPDO,IDUM2(2),NMT,IT11
      WRITE (6,12)
 12   FORMAT (        T12,'TIME', T21,'RPM' , T32,'PSI', T40,'BET-LG',
     2  T51,'BET-LT', T61,'BET(1)',                T72,'FFX', T82,'FFY',
     3  T92,'FFZ',T102,'WBX',T112,'WBY',T122,'WBZ'/ T12,'FI', T21,'FJ',
     4  T32,'FK', T41 ,'MI'  ,  T52 ,'MJ'  , T62 ,'MK'   , T72,'TFFX',
     5  T82,'TFFY', T92,'TFFZ',T102,'TFMX',T112,'TFMY',T122,'TFMZ' )
      IF (IT4.EQ.1) GO TO 26
      WRITE (6,14)
 14   FORMAT (T12,'1BM1', T21,'1CM1', T32,'2BM1', T40,'2CM1', T51,'3BM1'
     1  , T61,'3CM1', T72,'1BM2', T82,'1CM2', T92,'2BM2', T102,'2CM2',
     2  T112,'3BM2', T122,'3CM2')
       IF (NMT.GT.2) WRITE (6,16)
 16   FORMAT (T12,'1BM3', T21,'1CM3', T32,'2BM3', T43,'2CM3', T51,'3BM3'
     1  ,T61,'3CM3', T72,'1BM4', T82,'1CM4', T92,'2BM4', T102,'2CM4',
     2  T112,'3BM4', T122,'3CM4')
      WRITE (6,40)
 40   FORMAT ( T12,'ZTIP1', T21,'XTIP1', T32,'ZTIP2', T43,'XTIP2', T51,
     1  'ZTIP3', T61,'XTIP3', T72,'1OB1',T82,'2OB1',T92,'3OB1',T102,
     2  '1OB2',T112,'2OB2',T122,'3OB2')
      IF (NBM.GT.2) WRITE (6,44)
 44   FORMAT ( T12,'1OB3',T21,'2OB3',T32,'3OB3',T41,'1OB4',T51,'2OB4',
     1  T61,'3OB4')
 26   IF (IT5.EQ.1) WRITE (6,22)
 22   FORMAT (          T12,'THP', T21,'PSIP', T32,'XMB',T61,'YMB',T52,
     1  'ZMB',T62 ,'4BM', T72 ,'WCD', T82 ,'MTN',T92,'QM1', T102,'QM2',
     2  T112,'QM3', T122,'QM4')
      IF (IT11.EQ.1) WRITE (6,20)
 20   FORMAT (                  T12,'FXL', T22,'FZL', T32,'MXL', T41,'MYL',
     5  T52,'MZL', T62,'FXR', T72,'FZR', T82,'MXR', T92,'MYR', T102,
     6  'MZR')
      IF (IT10.EQ.0) RETURN
      WRITE (6,42)
 42   FORMAT (T12,'XPDO1', T21,'YPDO1', T32,'ZPDO1', T40,'XPDO2', T51,
     1  'YPDO2', T61,'ZPDO2', T72,'XPDO3', T92,'YPDO3', T92,'ZPDO3',
     2  T102,'XPDO4', T112,'YPDO4', T122,'ZPDO4')
      IF (NPDO.GT.4) WRITE (6,18)
 18   FORMAT (T12,'XPDO5', T21,'YPDO5', T32,'ZPDO5', T40,'XPDO6', T51,
     1  'YPDO6', T61,'ZPDO6', T72,'XPDO7', T92,'YPDO7', T92,'ZPDO7',
     2  T102,'XPDO8', T112,'YPDO8', T122,'ZPDO8')
      RETURN
      END
                              CARD COUNT        44
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE INPUT
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /FLAP6/ IT1,IT2,IT3,IT4,IT5,IT6,IT7,IT8,IT9,IT10,NS,NB,
     1  NPSTN,NGOTO,IDUM2(2),NBM,NB,NP,KPR(4),NP,NSW,NSHP,IDUM1(5),KP,
     2  KSN(4),KSNE(4),IT12,NOAIR,IDUM3(5),IT13,NPT,IT14
      COMMON /INPU/   AM(10),A0,A1,B1,THC,TDEL3,THT(10)
     2  , PI(10), YR(10), FPC(10), FPE(10), AIC(10), EPF(10), C(10)
     3  , AMG(10),S,J,ZH,RHO,RR(10),REALTM,OMF(4),AK(4),BK(4),DOMA(3),
     3  COMB(3),ALW,TCA,UI,VI,TL,
     4  P1,R2,R1SQ,R2SQ,ALL,WC,OMA,CPS,STOPK,STOPTH,DTMIN,AMPV,AMPH,
     5  THCMAX,THST,THEO,PTHA,PTHB,G,AMGY,GYM(10)
     1  ,        CTHBR(10),STHBR(10),AIMYS(10),AMY2(10),CTHT(10),STHT(10)
     2  ,OMAC(4),AIC(4),OMACS(4),TAIBI,SLC,DPSI,GST1,GED1,GST2,GFD2,
     3  GUST1,GUST2,CNB(3),SNB(3)
      COMMON /AIPLD/ DUM1(30),HRCDR(10),HRCSDR(10),DUM3(48),PHE(10,3),
     1  PI(10),TOC(10),DUM4(6),VGH,VGV
      COMMON / FBIN / BE(10,4),XI(10,4),YBT(10,4),THM(10,4),ZE(3,4,3),
     1  ZFF(3,4),BFM(4),BOMSQM(4),BOM2M2(4),AMB(10,4),AMYBT(10,4),
     3  AMX(10,4),AIT(10,4),DUM6(24),BOM(4),RZETA
      COMMON / WIN /                                  WY(10),WEPE,
     1          PHXA( 4),PHYA( 4),PHZA( 4),PHTA( 4),PHPA( 4),PHX(10, 4),
     2  PHZ(10, 4),PHT(10, 4),             WMG(10),HPCR(10),OMSQ( 4),ZM2( 4)
     3  ,CTPC,STPC,EM( 4),PHZY(10, 4),OOA( 4),TPC,SHPDE(4),SHPE(4),
     4 SHPD(4,2),DUM(9),OFFX(4),OFFY(4),OFFZ(4),DUM7(67),PHXY(10, 4),
     5  ALO,          SYMM,                          WRS,OME,OMES,ALX,ALS
      COMMON /ROWRT/ RPM,OMT,DEL3,WZETA,PM,ZP,XP,HM,WIPP,WM(10),WIE(10),
     1  WDR(10),OM( 4),PPMF(4),DEL, ALAM,DR(10),VP,GY,SPME,COF,PSIE
      COMMON /FLAP4/ DUM2(17),ATHOO( 9),DUMB(408),PKE(4)
      COMMON / FLEXDT/ ET(10,3,10),CB,SB
      COMMON /WINGT/ XY(15),DUM8(80),WT(10,3),DUM9(37),FRA(5)
      COMMON / FLXDT/ FA1X,FA2X
      DATA RAD/.017453/
C
      CALL READWR
      NOAIR = 0
      IF (RHO.EQ.0.00 .OR. RPM.EQ.0.00.AND.U.EQ.0.00) NOAIR = 1
      SNB(1) = 0.00
      SNB(2) = DSQRT(3.00)/2.00
      SNB(3) = -SNB(2)
      CNB(1) = 1.00
      CNB(2) = -.500
      CNB(3) = -.500
      DPSI = .523600
      IF (IT14.EQ.1) GO TO 12
      DPSI = DPSI/2.00
      DTMIN = DTMIN/2.00
12    NPT = REALTM/DTMIN/450.00 + .999900
      NPT =  MAX0(NPT,3)
      RNS = RNS)
      THC  = RAD*THC
      RHO = RHO/12.**4
      S = S*12.
      TPC = TPC*RAD
      CTPC =DCOS(TPC)
      STPC =DSIN(TPC)
```

ARAP06 FORTRAN Listings (Continued)

```
      DMA = RPM*.10472
      DO 15 K=1,15
   15 XY(K) = 0.D0
      DO 14 K=1,9
   14 ATMOD(K) = 0.D0
      XY(9) = TPC
      ATMOD(6) = DMA
      ATMOD(9) = PSIE*RAD
      A0    = RAD*A0
      CB = DCOS(A0)
      SB = DSIN(A0)
      TDEL3 =DTAN(DEL3*RAD)
      A1    = RAD*A1
      B1    = RAD*B1
      U = U*1.689*12.
      AMGV = 0.D0
      DO 50 I=1,NS
      AMGV = AMGV + AM(I)
      GVM(I) = AM(I)*GV
      RR(I) = R(I)/RVS
      AMG(I) = AM(I)*G
      AMV2(I) = 2.*AM(I)*YB(I)
      AIMYS(I) =AIC(I) + AM(I)*YB(I)*YB(I)
      THT(I) = THT(I)*RAD
      CTHT(I) =DCOS(THT(I))
      STHT(I) =DSIN(THT(I))
      HRCDR(I) = .5*RHO*C(I)*DR(I)
      HRCSDP(I)= .5*RHO*C(I)**2*DR(I)
      TOC(I) = 2./OMT/C(I)
      B(I) = RHO*OMT*C(I)**4*DR(I)
      DO 50 J=1,NB
      PHE(I,J) = 0.D2
      DO 50 K=1,10
      ET(I,J,K) = 0.D0
   50 CONTINUE
      AMGV = AMGV*GV*3.
      VGH = 0.D0
      VGV = 0.D0
      FA1K = 0.D0
      FA2K = 0.D0
      UI = 1.D0
      VI = 0.D0
      PTHA = 0.D0
      IF (IT2.EQ.1) GO TO 24
      THCMAX = THCMAX*RAD
      PTHA = (THCMAX - THC)/(TMED - TMST)
      PTHB = THC - PTHA*TMST
   24 IF (IT3.EQ.0) GO TO 26
      IF (GST1.NE.GED1) GUST1 = 1.D0/(GED1-GST1)
      IF (IT3.EQ.2) GUST2 = 3.14159625D0/(GED2 [illegible]2)
      CPS = CPS*6.283
   26 IF (IT1.EQ.0) GO TO 77
      STOPK = STOPK*687.6
      STOPTH = STOPTH*RAD
      DO 71 K=1,4
```

PAGE 3

```
      OMF(K) = RPMF(K)*.10472
      AK(K) = AK(K)*687.6
71    BK(K) = BK(K)*687.6
      DO 73 K=1,3
      DCMA(K) = 0.D0
      DCMB(K) = 0.D0
      IF (OMF(K).EQ.OMF(K+1)) GO TO 73
      DCPA(K) = (AK(K+1) - AK(K))/(OMF(K+1) - OMF(K))
      DOMB(K) = (BK(K+1) - BK(K))/(OMF(K+1) - OMF(K))
73    CONTINUE
77    IF (IT5.EQ.0 .AND. IT7.EQ.0) GO TO 95
      ALW = ALW*RAD
      UI =DCOS(ALW)
      VI =DSIN(ALW)
      ALO = UI
      IF (IT7.EQ.0) GO TO 32
      R1 = .25 + .25*DEL
      R2 = .25*DSQRT(1. - DEL*DEL)
      R1SQ = R1*R1
      R2SQ = R2*R2
      ALAM = ALAM*RAD
      TL = DSIN(ALAM)/DCOS(ALAM)
32    IF (IT5.EQ.0) GO TO 95
      CTPC2 = 2.*CTPC
      STPC2 = 2.*STPC
      ZPSQ = ZP*ZP
      PMG = PM*G
      HMG = HM*G
      GYMH = GY*HM
      DO 60 I=1,NSW
      WMG(I) = WM(I)*G
      MACR(I) = .5*RHO*WC*WDR(I)
      DO 60 J=1,3
      WT(I,J) = 0.D0
60    CONTINUE
      YP = -YP
      DO 75 J=1,NSHP
      ZMZ(J) = 2.*WZETA*OM(J)
      OMSQ(J) = OM(J)**2
      WEP = 0.D0
      PHPA(J) = -PHPA(J)
      PHZA(J) = -PHZA(J)
      PHXYZA = PHXA(J)**2 + PHYA(J)**2 + PHZA(J)**2
      PHPASQ = PHPA(J)**2
      PHTASQ = PHTA(J)**2
      PHTAXA = PHTA(J)*PHXA(J)
      PHYAPA = PHYA(J)*PHPA(J)
      PHZATA = PHZA(J)*PHTA(J)
      DO 65 I=1,NSW
      PHZ(I,J) = -PHZ(I,J)
      PHZV(I,J) = PHZ(I,J) + WY(I)*PHT(I,J)*UI
      PHXV(I,J) = PHX(I,J) - WY(I)*PHT(I,J)*VI
65    WEP = WEM + WM(I)*(PHX(I,J)**2 + PHZ(I,J)**2) + WIE(I)*PHT(I,J)**2
     1  + (PHZ(I,J)*UI - PHX(I,J)*VI)*PHT(I,J)*WM(I)*WY(I)*2.
      IF (IT13.EQ.0)
```

ARAP06 FORTRAN Listings (Continued)

```
      IFM(J) = WEM + PHM*(PHXYZA + (ZPSQ + YP*YP)*PHPASQ + (ZPSQ + XP*XP)
     1  *PHTASQ - PHTAXA*(XP*CTPC2 + ZP*STPC2) - YP*(-2.*XP*PHTA(J)*
     2  PHPA(J) + CTPC2*PHPA(J)*PHXA(J) + STPC2*PHZA(J)*PHPA(J)) - 2.*
     3  ZP*PHYAPA + PHZATA*(ZP*CTPC2 - XP*STPC2)) + HM*(PHXYZA + ZH*
     4  (ZH*(PHPASQ + PHTASQ) - STPC2*PHTAXA - 2.*PHYAPA *CTPC2*PHZATA))
     5  +    WIPP*(PHTASQ + PHPASQ)
      QOA(J)=         - PMG*(PHZA(J) + (ZP*CTPC - XP*STPC)*PHTA(J) - YP*
     1  STPC*PHPA(J))  - HMG*(PHZA(J) + ZH*CTPC* PHTA(J)) + (ZP*PHPA(J)
     2  - PHYA(J))*PM*GY
      QFFX(J) = ZH*PHTA(J) + CTPC*PHZA(J) - STPC*PHXA(J)
      QFFY(J) =PHYA(J) - ZH*PHPA(J)
      QFFZ(J) =  CTPC*PHXA(J) + STPC*PHZA(J)
      SHPDE(J) = 0.D0
      XY( 6) = PHXA(J)*SHPE(J) + XY( 6)
      XY( 7) = PHYA(J)*SHPE(J) + XY( 7)
      XY( 8) = PHZA(J)*SHPE(J) + XY( 8)
      XY( 9) = PHTA(J)*SHPE(J) + XY( 9)
      XY(10) = PHPA(J)*SHPE(J) + XY(10)
      DO 70 K=1,4
      SHPDD(J,K) = 0.D0
70    CONTINUE
95    IF (IT8.NE.1) GO TO 20
      DO 36 K=1,4
      OMAC(K) = OMAC(K)*6.2831D0
      OMACS(K) = OMACS(K)*6.2831D0
36    AIC(K) = AIC(K)*RAD
20    IF (IT8.NE.2) GO TO 34
      HRS = .5D0*RHO*SPHE*CDE
      OME = OME*6.283D0
      OMES = OMES*6.283D0
      ALX = ALX*RAD
      ALS = ALS*RAD
      DO 18 K=1,5
18    FMA(K) = 0.D0
34    IF (IT9.EQ.0) GO TO 30
      KP = 1
      IF (IT5.EQ.1) KP = 4
      DO 38 K=1,KP
      PKE(K) = 0.D0
      KPK(K) = 0
      KSN(K) = 0
38    KSNE(K) = 0
30    IF (IT4.EQ.1) RETURN
      CT =DCOS(THC)
      ST =DSIN(THC)
      DO 45 I=1,NS
      CTHBB(I) = CT*CTHT(I) - ST*STHT(I)
      STHBB(I) = ST*CTHT(I) + CT*STHT(I)
45    CONTINUE
      NGOTO = 0
      CALL TABINA(THC,OMA)
      NGOTO = -1
      CALL TABINA(THC,OMA)
      DO 205 N=1,NBM
      BEM( N) = 0.D0
```

ARAP06 FORTRAN Listings (Continued)

```
      DO 22  J=1,NB
      DO 22 K=1,3
22    ZF(J,N,K) = 0.D0
      DO 204 I=1,NS
      BFM(   N) = BEN(   N) + AM(I)*(BE(I,   N)**2 + XI(I,   N)**2) +
     1  AIMYS(I)*THM(I,   N)**2 + AMY2(I)*(CTHBR(I)*T4M(I,   N)*BE(I,   N)+
     2  STHBR(I)*THM(I,   N)*XI(I,   N))
      AMB(I,   N) = AM(I)*BE(I,   N)
      AMYBT(I,   N) = AM(I)*YB(I)*THM(I,N)
      AMX(I,   N) = AM(I)*XI(I,   N)
      YBT(I,   N) = YB(I)*THM(I,   N)
      AIT(I,N) = AIC   (I)*THM(I,N)
204   CONTINUE
      BOMSOM(   N) =           BOM(   N)**2
      BOMZM2(   N) = 2.*BZETA*          BOM(   N)
205   CONTINUE
      RETURN
      END
                          CARD COUNT        230
```

ARAP06 FORTRAN Listings (Continued)

```
                                                              PAGE    1
      SUBROUTINE PLTS
      REAL*8  DUM1(119),REALTM
      COMMON /INPU/ DUM1,REALTM
      COMMON /FLAP6/ IDUM1(5),IT6,IDUM2(11),NPTS,IDUM3(32),IBUF(512)
      DIMENSION            X(450),Y(450),Z(450),DUM(7),TITLE(20),
     1  L1(7,3),L2(7,3),T1(47),T2(60),T3(45)
      DATA T1/'RPM = ROTOR RPM BET-LG = LONGITUDINAL FLAPPING  BET-LT =
     1LATERAL FLAPPING   TFFX = ROTOR TOTAL H-FORCE  TFFY = ROTOR TOTAL
     2Y-FORCE  TFFZ = ROTOR TOTAL THRUST   TFMZ = ROTOR TOTAL TORQUE   '
     3/, T2/'D1 = MASTER BLADE TIP OUT-OF-PLANE DEF  D2 = MASTER BLADE T
     4IP INPLANE DEF   M1 = 1ST BEAMWISE BENDING MOMENTM2 = 1ST CHORDWIS
     5E BENDING MOMENT   M3 = 2ND BEAMWISE BENDING MOMENTM4 = 2ND CHORDW
     6ISE BENDING MOMENT   B1 = MASTER BLADE FLAPPING  '/,
     7  T3/'WBM = WING TIP BEAMWISE DEF WCD = WING TIP CHORDWISE DEFWTN
     8= WING TIP TORSIONAL DEFXPDD1 = 1ST VERTICAL ACCYPDD1 = 1ST LATERA
     9L ACC ZPDD1 = 1ST AXIAL ACC   XPDD2 = 2ND VERTICAL ACC'/,
     X  L1/16,32,5*28, 40,36,32,36,32,36,28, 3*28,4*24/,
     1  L2/1,5,13,20,27,34,41, 1,11,20,28,37,45,54, 1,8,15,22,28,34,40/
C
      REWIND 1
      READ (1) TITLE
      NPTS1 = NPTS   + 1
      NPTS2 = NPTS   + 2
      L = REALTM*2. + 1.001
      AX = L
      PX = AX + 4.
      DO 10 I=1,NPTS
   10 READ (1) X(I),     (DUM(K),K=1,7)
      CALL PLOT (0.,2.,-3)
      X(NPTS1) = .0
      X(NPTS2) = .5
      CALL SYMBOL (0.,6.4,.21,TITLE,0.,80)
      YPEN = 5.5
      DO 24 K=1,7
      L3 = L1(K,IT6)
      YPEN = YPEN - .5
      L4 = L2(K,IT6)
      GO TO (11,15,17), IT6
   11 CALL SYMBOL (0.,YPEN,.14,T1(L4),0.,L3)
      GO TO 24
   15 CALL SYMBOL (0.,YPEN,.14,T2(L4),0.,L3)
      GO TO 24
   17 CALL SYMBOL (0.,YPEN,.14,T3(L4),0.,L3)
   24 CONTINUE
      CALL PLOT (6.   ,0.,-3)
      DO 50 KL=1,4
      REWIND 1
      READ (1) TITLE
      KGO = (IT6-1)*4 + KL
      GO TO (12,14,16,26), KL
   12 DO 13 I=1,NPTS
   13 READ (1)  DUM(1),        Y(I),Z(I),(DUM(J),J=1,5)
      GO TO 18
   14 DO 20 I=1,NPTS
   20 READ (1) (DUM(K),K=1,3),Y(I),Z(I),(DUM(J),J=1,3)
```

ARAP06 FORTRAN Listings (Continued)

```
      GO TO 18
16    DO 22 I=1,NPTS
22    READ (1) (DUM(K),K=1,5),Y(I),Z(I),DUM(1)
      GO TO 18
26    DO 28 I=1,NPTS
28    READ (1) (DUM(K),K=1,7),Y(I)
18    IF (KL.NE.1) CALL PLOT (PX,0.,-3)
      CALL SCALE (Y, 5.,NPTS,1,20.0)
      CALL AXIS (0.,0.,10HTIME (SEC),-10, AX,0.,0.,.5,20.)
      GO TO (31,32,33,34,35,36,37,39,51,52,53,54), KGO
31    CALL AXIS (0.,0.,3HRPM,3,5.,90.,Y(NPTS1),Y(NPTS2),20.)
      GO TO 38
32    CALL AXIS (0.,0.,12HBET-LT (DEG),12,5.,90.,Y(NPTS1),Y(NPTS2),20.)
      GO TO 38
33    CALL AXIS (0.,0.,10HTFFY (LBS),10,5.,90.,Y(NPTS1),Y(NPTS2),20.)
      GO TO 38
34    CALL AXIS (0.,0.,12HTFMZ (IN-LB),12,5.,90.,Y(NPTS1),Y(NPTS2),20.)
      GO TO 38
35    CALL AXIS (0.,0.,7HD1 (IN),7,     5.,90.,Y(NPTS1),Y(NPTS2),20.)
      GO TO 38
36    CALL AXIS (0.,0.,10HM1 (IN-LB),10,  5.,90.,Y(NPTS1),Y(NPTS2),20.)
      GO TO 38
37    CALL AXIS (0.,0.,10HM3 (IN-LB),10,  5.,90.,Y(NPTS1),Y(NPTS2),20.)
      GO TO 38
39    CALL AXIS (0.,0.,8HB1 (DEG),8,      5.,90.,Y(NPTS1),Y(NPTS2),20.)
      GO TO 38
51    CALL AXIS (0.,0., 8HWBM (IN),8    ,5.,90.,Y(NPTS1),Y(NPTS2),20.)
      GO TO 38
52    CALL AXIS (0.,0.,9HMTM (RAD),9,5.,90.,Y(NPTS1),Y(NPTS2),20.)
      GO TO 38
53    CALL AXIS (0.,0.,9HYPDD1 (G),9,5.,90.,Y(NPTS1),Y(NPTS2),20.)
      GO TO 38
54    CALL AXIS (0.,0.,9HXPDD2 (G),9,5.,90.,Y(NPTS1),Y(NPTS2),20.)
38    CALL LINE (X,Y,NPTS,1,0,0)
      IF (KL.EQ.4) GO TO 50
      CALL PLOT (PX ,0.,-3)
      CALL SCALE (Z, 5.,NPTS,1,20.0)
      CALL AXIS (0.,0.,10HTIME (SEC),-10, AX,0.,0.,.5,20.)
      GO TO (41,42,43,50,45,46,47,50,61,62,63,50), KGO
41    CALL AXIS (0.,0.,12HBET-LG (DEG),12,5.,90.,Z(NPTS1),Z(NPTS2),20.)
      GO TO 48
42    CALL AXIS (0.,0.,10HTFFX (LBS),10,5.,90.,Z(NPTS1),Z(NPTS2),20.)
      GO TO 48
43    CALL AXIS (0.,0.,10HTFFZ (LBS),10,5.,90.,Z(NPTS1),Z(NPTS2),20.)
      GO TO 48
45    CALL AXIS (0.,0.,7HD2 (IN),7,     5.,90.,Z(NPTS1),Z(NPTS2),20.)
      GO TO 48
46    CALL AXIS (0.,0.,10HM2 (IN-LB),10,  5.,90.,Z(NPTS1),Z(NPTS2),20.)
      GO TO 48
47    CALL AXIS (0.,0.,10HM4 (IN-LB),10,  5.,90.,Z(NPTS1),Z(NPTS2),20.)
      GO TO 48
61    CALL AXIS (0.,0., 8HWCD (IN),8    ,5.,90.,Z(NPTS1),Z(NPTS2),20.)
      GO TO 48
62    CALL AXIS (0.,0., 9HXPDD1 (G),9,    5.,90.,Z(NPTS1),Z(NPTS2),20.)
      GO TO 48
```

ARAP06 FORTRAN Listings (Continued)

```
63    CALL AXIS (0.,0., 9HZPDD1 (G),9,   5.,90.,Z(NPTS1),Z(NPTS2),20.)
48    CALL LINE (X,Z,NPTS,1,0,0)
50    CONTINUE
      CALL PLOT (PX ,-2.,-3)
      RETURN
      END
                          CARD COUNT      116
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE PSLOP (A, DA, I)
      REAL *8 A(1),DA(1),AG,DAB,FUN,A1S,A2S,A3S,AC,BC,CC
     1       ,A3,A4,A5,A6,A7,A8,A9,A1,A2
      FUN(A1,A2,A3,A4,A5,A6,A7,A8,A9) = A3*(A4*A8 - A5*A7) + A1*(A5*A9 -
     1  A6*A8) - A2*(A4*A9 - A6*A7)
      IF (I-2) 1,4,8
    8 A1S = A(1)**2
      A2S = A(2)**2
      A3S = A(3)**2
      AC =FUN(DA(1),A(1),1.D0, DA(2),A(2),1.D0,DA(3),A(3),1.D0)
      BC =FUN(A1S,DA(1),1.D0,A2S,DA(2),1.D0,A3S,DA(3),1.D0)
      CC =FUN(A1S,A(1),DA(1),A2S,A(2),DA(2),A3S,A(3),DA(3))
      IF (AC.NE.0.D0) GO TO 6
      IF (BC.EQ.0.D0) GO TO 7
      AG = -CC/BC
      GO TO 5
    6 AG = -BC/AC*.5D0
      DAB = AG*AG - CC/AC
      IF (DAB.LT.0.D0) GO TO 4
      DAB = DSQRT(DAB)
      AG=AG-DAB*DSIGN(1.D0,DAB*(AG-A(1)))
      GO TO 5
    4 DAB = DA(2) -DA(1)
      IF (DABS(DAB).LT.1.D-10) GO TO 3
      AG =A(1)- DA(1)*(A(2)- A(1))/ DAB
      GO TO 5
    1 AG = A(1) +DA(1)*.5
      GO TO 2
    3 AG=(A(1)+A(2))*.5
    5 DA(3)=DA(2)
      A(3) =A(2)
    2 DA(2) = DA(1)
      A(2) = A(1)
      A(1) = AG
    7 RETURN
      END
                        CARD COUNT        36
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE READWR
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 TITLE(20)
      COMMON /FLAPW/ IT1,IT2,IT3,IT4,IT5,IT6,IT7,IT8,IT9,IT10,NS,NB,
     1  IDUM3(4),NRM,IDUM1(7),NSW,NSHP,NPOD,NTHC,NPSID,NMT,
     2  IT11,IDUM4(4),IT12,NOAIR,NC,IDUM2(4),IT13,NPT,IT14
      COMMON /INPU/   AM(10),A0,A1,B1,THC,TDEL3,THT(10)
     2  , R(10), YB(10), EPC(10), EPE(10), AIC(10), EPF(10), C(10)
     3  ,ANG(10)  ,S,U,ZM,            RHO,  RR(10),REALTM,DUM6(4),
     7                                 AK11,AK12,AK13,AK14,BK11,BK12,
     8  BK13,BK14,DUM5(6),ALW,YCA,DUM6(7),AL,WC,OMA,CPS,STOPK,STOPTM,
     5  DTMIN,AMPV,AMPH,THCMAX,TMST,TMED,PTHA,PTHB,G,DUM3(71),OMAC,OMAS,
     1  OMBC,OMBS,A1C,A1S,B1C,B1S,OMACS,OMASS,OMBCS,OMBSS,TA(8),SLC
     2  ,DPSI,GST1,GED1,GST2,GED2
      COMMON / FRIN / DUM1(408),BZETA
      COMMON / WIN / WY(10),WEPE,PHXA(4),PHYA(4),PHZA(4),PHTA(4),
     1  PHPA(4),PHX(10,4),PHZ(10,4),PHT(10,4),DUM9(33),EM(4),
     2                          DUMP( 44),TPC,SHPDE(4),SHPE(4),DUM(29),
     3  PX(9),PY(9),PZ(9),RMTRX(10,4),DUM7(43),OME,OMES,ALX,ALS,UE,VE,WE
     4  ,TEX,CLXO
      COMMON /ROWRT /        RPM,OMT,DEL3,WZETA,PM,ZP,XP,HM,WIPP,WM(10),
     1  WIE(10),WOR(10),DM( 4),RPM1,RPM2,RPM3,RPM4,DEL,
     2  ALAM,DR(10),YP,GY,SPNE,CDE,PSIE
       DIMENSION                                  NPF(35),MTX(5),NSP(15),
     1             NPHX(14),NPHZ(14),NPHT(14),NFT(10)
      DATA NPHX /'(/17H PHX(I,J) (IN/IN),3X,      10F11.5 /(20X,    7F11.5
     1 )'/,NPHZ /'(/17H PHZ(I,J) (IN/IN),3X,      10F11.5 /(20X,    7F11.5
     2 )'/,NPHT /'(/18H PHT(I,J) (RAD/IN),2X,     10F11.5 /(20X,    7F11.5
     3 )'/,NFT/'    1    2    3    4    5    6    7    8    9   10'/,NPF/'(//24H A
     1CCELEROMETER LOCATIONS //' 8X,1HK,4X,   9I12 // 5X,8HPX(K) IN,2X,
     2 9F12.3//5X,8HPY(K) IN,2X,   9F12.3//5X,8HPZ(K) IN,2X,   9F12.3 /)
     2'/,MTX /'(10X,        5D15.4  )'/,NSP/'(/6H SHPE(,I1,3H) =,F7.3,4X,
     3  4(5HSHPE(,I1,3H) =,F7.3,6X))'/ ,NCA/0/
C
10    FORMAT (16I5)
12    FORMAT (20A4)
14    FORMAT (5D15.0,5X)
15    FORMAT (8F10.0)
16    FORMAT (// ' WING STIFFNESS MATRIX' / )
20    FORMAT (1H1, T38,'ARAP06  FOLDING PROPROTOR FEATHER-STOP DYNAMIC A
     1NALYSIS' //// ' TITLE',10X,20A4 //// ' FLIGHT CONDITION' // ' U ='
     2  ,F7.2,' KTS', T29,'RPM =',F6.1, T54,'G =',F6.1,' IN/SEC**2',
     3  T80,'GY =',F6.1,' IN/SEC**2', T106,'S =',F6.0,' FT/SEC' //
     4  ' TPC =',F6.2,' DEG', T28,'RHO =',F9.6,' SLUG/FT**3'///' OPTIONS
     5 AND CONTROLS' // ' IT1 =',I2, T28,'IT2 =',I2, T54,'IT3 =',I2,T80,
     6  'IT4 =',I2, T106,'IT5 =',I2 // ' IT6 =',I2, T28,'IT7 =',I2, T54,
     7  'IT8 =',I2, T80,'IT9 =',I2, T106,'IT10 =',I2 // ' IT11 =',I2,
     8  T28,'IT12 =',I2, T54,'IT13 =',I2, T80,'IT14 =',I2, T106,'IT15 ='
     9  ,I2 // ' NCASE =',I2, T28,'DTMIN =',F7.4,' SEC', T54,'PSIE =',
     X  F6.1,' DEG')
22    FORMAT (/ ' THCMAX =',F7.2,' DEG',  T28,'TMST =',F5.2,' SEC',
     1  T54,'TMED =',F5.2,' SEC')
25    FORMAT ( // ' REAL TIME LIMIT =',F6.2,' SEC'
     2          //// ' END OF INPUT DATA')
42    FORMAT (/ ' OMAC =',F6.2,' CPS', T20,'OMAS =',F6.2,' CPS', T54,
```

ARAP06 FORTRAN Listings (Continued)

```
     1 'OMRC =',F6.2,' CPS',T80,'OMBS =',F6.2,' CPS', T106,'AIC =',F6.2
     2 ,' DEG' // ' AIS =',F6.2,' DEG', T28,'BIC =',F6.2,' DEG', T54,
     3 'BIS =',F6.2,' DEG', T80,'OMACS =',F6.2,' CPS/SEC',  T106,
     4 'OMASS =',F6.2,' CPS/SEC' // ' OMBCS =',F6.2,' CPS/SEC',  T28,
     5 'OMBSS =',F6.2,' CPS/SEC', T54,'TA181 =',F5.2,' SEC')
45    FORMAT (// ' PROPROTOR DATA' // ' NS =',I3, T28,'AD =',F6.1,' DEG'
     1 , T54,'DEL3 =',F7.2,' DEG', T80,'THC =',F6.2,' DEG', T105,'AI ='
     2 ,F6.2,' DEG' // ' BI =',F6.2,' DEG', T28,'NBM =',I2, T54,
     3 'BZETA =',F6.3, T80,'OMT =',F7.2,' RAD/SEC', T106,'ZM =',F6.1,
     4 ' IN' // ' NMT =',I2, T28,'NTHC =',I2, T54,'NPSID =',I2
     1      /// ' SEGMENT   THI(I)       R(I)       AM(I)       YB(I)
     7     AIC(I)       EPF(I)       EPC(I)       EPE(I)       C(I)' ,
     8 7X, 'DR(I)' / 12X
     9 , 'DEG',10X, 'IN', 8X,'CHUG', 10X,'IN', 6X,'CHUG-IN.SQ', 7X,'IN',
     X 10X,'IN', 10X, 'IN', 10X,'IN' ,10X,'IN'//  (I5, F11.2,F12.1,
     1 F12.4, F12.2, F13.4, 4F12.2, F12.1))
50    FORMAT (/ ' AMPV =',F6.1,' IN/SEC', T28,'AMPH =',F5.1,' IN/SEC',
     1 T54,'GST2 =',F5.2,' SEC', T80,'GED2 =',F5.2,' SEC')
52    FORMAT (/ ' UE =',F6.2,' IN',  T28,'VE =',F6.2,' IN',  T54,'WE =',
     1 F6.2,' IN',  T80,'SPNE =',F6.2,' IN',  T106,'CDE =',F6.2,' IN'
     2 // ' OME =',F6.2,' CPS',  T28,'ALX =',F6.2,' DEG',  T54,'ALS =',
     3 F6.2,' DEG',  T80,'TEX =',F5.2,' SEC',  T106,'CLXO =',F6.3//
     4 ' OMES =',F6.2,' CPS/SEC')
55    FORMAT (                    / ' RPM1 =',F6.1,' RPM',  T28,'RPM2 =',
     6 F6.1,' RPM',  T54,'RPM3 =',F6.1,' RPM',  T80,'RPM4 =',F6.1,
     7 ' RPM',  T105,'AK11 =',F8.1,' FT-LB/DEG' //  ' AK12 =',F8.1,
     8 ' FT-LB/DEG',  T28,'AK13 =',F8.1,' FT-LB/DEG',  T54,'AK14 =',
     9 F8.1,' FT-LB/DEG',  T80,'BK11 =',F8.1,' FT-LB/DEG',  T105,
     X 'BK12 =',F8.1,' FT-LB/DEG'  //  ' BK13 =',F8.1,' FT-LB/DEG',
     1 T28,'BK14 =',F8.1,' FT-LB/DEG', T54,'STOPK =',F8.1,' FT-LB/DEG',
     2 T80,'STOPTH =',F6.1,' DEG')
56    FORMAT (/' AMPV =',F6.1,' IN/SEC', T28,'AMPH =',F6.1,' IN/SEC',
     1 T54,'GST1 =',F5.2,' SEC', T80,'GED1 =',F5.2,' SEC')
60    FORMAT (/ ' AMPV =',F6.1,' IN/SEC',  T28,'AMPH =',F6.1,' IN/SEC',
     1 T54,'CPS =',F6.2)
65    FORMAT (/' ALW =',
     1 F7.2,' DEG',  T28,'WC =',F7.2,' IN', T54,'SLC =',F6.3)
105   FORMAT ( /                      ' DEL =',F5.2,  T28,'ALAM =',F7.2,
     2 ' DEG',  T54,'AL =',F6.2,' IN',  T80,'YCA =',F7.2,' IN'   )
110   FORMAT (/////' WING/PYLON DATA'       /                /  ' NSW =',I3,
     1T22,'NSHP =',I3,T44,'WZETA =',F6.3, T66,'PM =',F5.2,' CHUG',  T88,
     2 'ZP =',F6.2,' IN',  T110,'YP =',F5.2,' IN' //  ' XP =',F5.2,
     3 ' IN', T22,'NPDD =',I2,                T44,'MM =',F5.2,' CHUG',  T66,
     4 'WEPE =',F5.2,' IN' ,T88, 'WIPP =',F7.0,' CHUG-IN**2')
115   FORMAT (//' SEGMENT(I)',4X, 10I11)
134   FORMAT ( //  T28,'J', T37,'OM(J)', T48,'PHXA(J)', T60,'PHYA(J)',
     1 T72,'PHZA(J)', T84,'PHPA(J)', T96,'PHTA(J)' / T35,'RAD/SEC',
     2 T49,'IN/IN',  T61,'IN/IN',  T73,'IN/IN',  T84,'RAD/IN',  T96,
     3 'RAD/IN'                 // (25X,I3,F14.3,5F12.5))
135   FORMAT ( //  T28,'J', T37,'OM(J)', T48,'PHXA(J)', T60,'PHYA(J)',
     1 T72,'PHZA(J)', T84,'PHPA(J)', T96,'PHTA(J)' , T108,'EM(J)' /
     1 T36,'RAD/SEC',
     2 T49,'IN/IN',  T61,'IN/IN',  T73,'IN/IN',  T84,'RAD/IN',  T96,
     3 'RAD/IN' ,T108,'CHUG' // (25X,I3,F14.3,5F12.5,F12.4))
140   FORMAT (/' WM(I)  (CHUG)',5X, 10F11.4)
```

ARAP06 FORTRAN Listings (Continued)

```
145   FORMAT (/' WIE(I) (CHUG-IN**2)', F10.4,9F11.4)
150   FORMAT (/' WY(I)  (IN)',5X, 10F11.2)
155   FORMAT (/' WDR(I) (IN)',5X, 10F11.2)
C
      NB = 3
      READ (5,12,END=26,ERR=26) TITLE
      READ (5,10) IT1,IT2,IT3,IT4,IT5,IT6,IT7,IT8,IT9,IT10,IT11,IT12,
     1  IT13,IT14,IT15
      READ (5,10) NS,NBM,NMT,NTHC,NPSID,NCASE
      IF (NCA.EQ.0) NC = NCASE
      NCA = 1
      READ (5,15) ZM,S,RHO,TPC,BZETA,RPM,OMT,G,AO,DEL3,THC,A1,B1,U,
     1  REALTM,GY,DTMIN,PSIE
      READ (5,15)   RPM1,RPM2,RPM3,RPM4,AK11,AK12,AK13,
     1  AK14,BK11,BK12,BK13,BK14,STOPK,STOPTH
      READ (5,15)  THCMAX,TMST,TMED
      READ (5,15) AMPV,AMPH,CPS,GST1,GED1,GST2,GED2
      READ (5,15) ALW,WC,SLC,DEL,ALAP,AL,YCA
      READ (5,15) OMAC,OMAS,OMBC,OMBS,AIC,AIS,BIC,BIS,OMACS,OMASS,OMBCS,
     1  OMBSS,TAIB1
      READ (5,15) UE,VE,WE,SPNE,CDE,OME,ALX,ALS,TEX,CLXO,OMES
      READ (5,15) (THT(I),I=1,NS)
      READ (5,15) (R(I)  ,I=1,NS)
      READ (5,15) (AM(I) ,I=1,NS)
      READ (5,15) (YB(I) ,I=1,NS)
      READ (5,15) (AIC(I),I=1,NS)
      READ (5,15) (EPF(I),I=1,NS)
      READ (5,15) (EPC(I),I=1,NS)
      READ (5,15) (EPE(I),I=1,NS)
      READ (5,15) (C(I)  ,I=1,NS)
      READ (5,15) (DR(I) ,I=1,NS)
      IF (DTMIN.EQ.0.D0) DTMIN = .5D-2
      IF (IT4.EQ.0) GO TO 40
      REWIND 1
      WRITE (1) TITLE
40    WRITE (6,20) TITLE,U,RPM,G,GY,S,TPC,RHO,IT1,IT2,IT3,IT4,IT5,IT6,
     1  IT7,IT8,IT9,IT10,IT11,IT12,IT13,IT14,IT15,NCASE,DTMIN,PSIE
      IF (IT1.EQ.1) WRITE (6,55) RPM1,RPM2,RPM3,RPM4,AK11,AK12,AK13,
     1  AK14,BK11,BK12,BK13,BK14,STOPK,STOPTH
      IF (IT2.EQ.0) WRITE (6,22) THCMAX,TMST,TMED
      IF (IT3.EQ.0) GO TO 36
      IF (IT3-2) 28,46,54
28    WRITE (6,56) AMPV,AMPH,GST1,GED1
      GO TO 36
46    WRITE (6,50) AMPV,AMPH,GST2,GED2
      GO TO 36
54    WRITE (6,60) AMPV,AMPH,CPS
36    IF (IT5.EQ.1 .OR. IT7.EQ.1) WRITE (6,65) ALW,WC,SLC
      IF (IT7.EQ.1) WRITE (6,105) DEL,ALAM,AL,YCA
      IF (IT8.EQ.1) WRITE (6,42) OMAC,OMAS,OMBC,OMBS,AIC,AIS,BIC,BIS,
     1  OMACS,OMASS,OMBCS,OMBSS,TAIB1
      IF (IT8.EQ.2) WRITE (6,52) UE,VE,WE,SPNF,CDF,OME,ALX,ALS,TEX,CLXO,
     1  OMES
      WRITE (6,45) NS,AO,DEL3,THC,A1,B1,NBM,BZETA,OMT,ZM,NMT,NTHC,
     1  NPSID,(I,THT(I),R(I),AM(I),YB(I),AIC(I),EPF(I),EPC(I),EPE(I),
```

ARAP06 FORTRAN Listings (Continued)

PAGE 4

```
2  C(I),DP(I),I=1,NS)
   IF (IT4.EQ.0) CALL BEXION
   IF (IT5.EQ.0) GO TO 90
   READ (5,17) NSW,NSHP,NPDD
   READ (5, 15) WZETA,PM,ZP,YP,XP,HM,WEPE,WIPP,(SHPE(J),J=1,NSHP)
   READ (5, 15) ((PHX(I,J),I=1,NSW),J=1,NSHP)
   READ (5, 15) ((PHZ(I,J),I=1,NSW),J=1,NSHP)
   READ (5, 15) ((PHT(I,J),I=1,NSW),J=1,NSHP)
   READ (5, 15) (WM(I),I=1,NSW),(WIF(I),I=1,NSW),(WY(I),I=1,NSW),
1  (WDR(I),I=1,NSW)
   READ (5, 15) (OM(J),J=1,NSHP),(PHXA(J),J=1,NSHP),(PHYA(J),J=1,
1  NSHP),(PHZA(J),J=1,NSHP),(PHPA(J),J=1,NSHP),(PHTA(J),J=1,NSHP)
1  ,(EM(J),J=1,NSHP)
   READ (5,14) ((RMTRX(I,J),I=1,10),J=1,NSHP)
   NPDD = MAX0(1,NPDD)
   READ (5,15) (PX(K),PY(K),PZ(K),K=1,NPDD)
   WRITE (6,110) NSW,NSHP,WZETA,PM,ZP,YP,XP,NPDD,HM,WEPE,WIPP
   IF (NSHP.GT.1) NSP(8) = NFT(NSHP-1)
   WRITE (6,NSP) (J,SHPE(J),J=1,NSHP)
   NPHX( 9)=NFT(NSW)
   NPHX(12)=NFT(NSW)
   NPHZ( 9)=NFT(NSW)
   NPHZ(12)=NFT(NSW)
   NPHT( 9)=NFT(NSW)
   NPHT(12)=NFT(NSW)
   WRITE (6,115) (I,I=1,NSW)
   WRITE (6,NPHX) ((PHX(I,J),I=1,NSW),J=1,NSHP)
   WRITE (6,NPHZ) ((PHZ(I,J),I=1,NSW),J=1,NSHP)
   WRITE (6,NPHT) ((PHT(I,J),I=1,NSW),J=1,NSHP)
   WRITE (6,140) (WM(I),I=1,NSW)
   WRITE (6,145) (WIE(I),I=1,NSW)
   WRITE (6,150) (WY(I),I=1,NSW)
   WRITE (6,155) (WDR(I),I=1,NSW)
   IF (IT13.EQ.0) WRITE (6,134) (J,OM(J),PHXA(J),PHYA(J),PHZA(J),
1  PHPA(J),PHTA(J),J=1,NSHP)
   IF (IT13.EQ.1) WRITE (6,135) (J,OM(J),PHXA(J),PHYA(J),PHZA(J),
1  PHPA(J),PHTA(J),EM(J),J=1,NSHP)
   IF (IT11.EQ.0) GO TO 44
   MTX(3) = NFT(NSHP)
   WRITE (6,16)
   WRITE (6,MTX)((RMTRX(I,J),J=1,NSHP),I=1,10)
44 IF (IT10.EQ.0) GO TO 90
   NPF(12) = NFT(NPDD)
   NPF(19) = NFT(NPDD)
   NPF(26) = NFT(NPDD)
   NPF(33) = NFT(NPDD)
   WRITE (6,NPF) (K,K=1,NPDD),(PX(K),K=1,NPDD),(PY(K),K=1,NPDD),
1  (PZ(K),K=1,NPDD)
90 WRITE (6,25) REALTM
   RETURN
26 STOP
   END
                  CARD COUNT      217
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE RSTN
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /FLAP6/IDUMA(11),NR,NRSTN
      COMMON /INPU/ DUM(120),OMF(4),AK(4),BK(4),DOMA(3),DOMB(3),
     1              DUMB(13),STOPK,STOPTH,DUM9(101),CNB(3),SNB(3)
      COMMON /AIPLD/ DUM1(29),CPSI
      COMMON /FLAP1/ DUM2( 99),BET(3),DUM7( 74),SPSI,DUM3(3),BLG,BLT
      COMMON /FLAP4/ DUM4( 6),PSID,DUM5(20),BBN5,CCN6
      COMMON /MIN/   DUM6(249),TFMX,TFMY
C
      DO 70 J=1,NB
      IF (BET(J).GE.STOPTH) GO TO 70
      XYM = (BET(J) - STOPTH)*STOPK
      XM = -SNB(J)*XYM
      YM =  CNB(J)*XYM
      IF (NRSTN.EQ.1) GO TO 76
      TFMX = TFMX - XM*CPSI + YM*SPSI
      TFMY = TFMY - XM*SPSI - YM*CPSI
      GO TO 74
76    BBN5 = BBN5 + XM
      CCN6 = CCN6 + YM
      GO TO 74
70    CONTINUE
      IF (PSID.LT.OMF(1)) GO TO 10
      AK1 = AK(1)
      BK1 = BK(1)
      GO TO 50
10    DO 20 I=1,3
      IF (PSID.LT.OMF(I+1)) GO TO 20
      AK1 = AK(I) + (PSID-OMF(I))*DOMA(I)
      BK1 = BK(I) + (PSID-OMF(I))*DOMB(I)
      GO TO 50
20    CONTINUE
      AK1 = AK(4)
      BK1 = BK(4)
50    XM = -BK1*BLT
      YM =  AK1*BLG
      IF (NRSTN.EQ.1) GO TO 60
      TFMX = TFMX - XM
      TFMY = TFMY - YM
      GO TO 74
60    BBN5 = BBN5 + XM*CPSI + YM*SPSI
      CCN6 = CCN6 - XM*SPSI + YM*CPSI
74    RETURN
      END
                              CARD COUNT        45
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE TABINA (THCH,PSIDH)
      REAL*8 BE,XI,BOM,DUM4,THM,YBT,THCH,PSIDH,TBMT,TCMT,BZETA
      COMMON /FBIN/ BE(10,4),XI(10,4),YBT(10,4),THM(10,4),DUM4(244),
     1  BOM(4),BZETA,TBMT(4,4),TCMT(4,4)
      COMMON /FBIN1/ FTHC(6),FPSID(6),OMG(144),DFTHC(5),DFPSID(5),
     1  FBE(1440),FXI(1440),FTH(1440),BMT(576),CMT(576)
      COMMON /FLAP6/ IDUM1(10),NS,NB,NRSTN,NGOTO,IDUM2(2),NBM,
     1  IDUM3(10),NTHC,NPSID,NMT
C
      PSID = PSIDH
      THC = THCH*57.296
      PSID = AMAX1(PSID,FPSID(1))
      PSID = AMIN1(PSID,FPSID(NPSID))
      THC = AMAX1(THC,FTHC(1))
      THC = AMIN1(THC,FTHC(NTHC))
      DO 10 K=2,NTHC
      IF (THC.LT.FTHC(K)) GO TO 20
   10 CONTINUE
      K = NTHC
   20 DX = (THC - FTHC(K-1)) / DFTHC(K-1)
      DO 30 M=2,NPSID
      IF (PSID.LT.FPSID(M)) GO TO 49
   30 CONTINUE
      M = NPSID
   40 DZ = (PSID - FPSID(M-1)) / DFPSID(M-1)
      IF (NGOTO.LT.3) GO TO 52
      KC = 6*M + K - 49
      KD = 60*M  + 10*K  - 500
      DO 50 N=1,NBM
      KA = 36*N + KC
      YLB = OMG(KA) + (OMG(KA+1)      - OMG(KA))*DX
      KA = KA+6
      BCM(N) = (OMG(KA) + (OMG(KA+1     ) - OMG(KA))*DX - YLB)*DZ + YLB
      DO 50 I=1,NS
      KA = 360*N + I + KD
      YLB = FBE(KA) + (FBE(KA+10) - FBE(KA))*DX
      YLX = FXI(KA) + (FXI(KA+10) - FXI(KA))*DX
      YLT = FTH(KA) + (FTH(KA+10) - FTH(KA))*DX
      KA = KA + 60
      BE(I,N) = (FBE(KA) + (FBE(KA+10) - FBE(KA))*DX - YLB)*DZ + YLB
      XI(I,N) = (FXI(KA) + (FXI(KA+10) - FXI(KA))*DX - YLX)*DZ + YLX
      THM(I,N)= (FTH(KA) + (FTH(KA+10) - FTH(KA))*DX - YLT)*DZ + YLT
   50 CONTINUE
      RETURN
   52 KC = 6*M + K - 193
      DO 60 N=1,NBM
      DO 60 L=1,NMT
      KA = 144*N + 36*L + KC
      YLB = BMT(KA) + (BMT(KA+1)      - BMT(KA))*DX
      YLC = CMT(KA) + (CMT(KA+1)      - CMT(KA))*DX
      KA = KA+6
      TBMT(L,N) = (BMT(KA) + (BMT(KA+1) - BMT(KA))*DX - YLB)*DZ + YLB
      TCMT(L,N) = (CMT(KA) + (CMT(KA+1) - CMT(KA))*DX - YLC)*DZ + YLC
   60 CONTINUE
      RETURN
```

ARAP06 FORTRAN Listings (Continued)

PAGE 2

```
      END
                        CARD COUNT       56
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE TABINT (V,N)
      COMMON /LLDD/ X,Z,CURVE(2688)
      DIMENSION  DSX(10,10),ILT(10),DSZ(31,10),NZ(10),NX(10),IZT(10)
C
      DATA  NX / 8,8,9,11,11,10,11,11,10,9/, NZ / 17,16,17,2*22,30,3*25,
     1 15/, ILT/ 0,161,313,492,887,1287,1622,1933,2244,2529/,DSX / .26,
     2 .04,.1,
     3 .1,.11,.05,.34,3*0.,  .26,.04,.1,.1,.11,.05,.34,3*0.,  .251,.151
     4 ,2*.101,.052,.054,.062,.228,2*0.,  .3,3*.1,.05,2*.05,.06,.03,.15,
     5 .3,3*.1,.06,2*.05,.06,.03,.15,  .301,.1,2*.101,2*.051,.053,.037
     6 ,.225,0.,  .3,3*.1,.06,3*.05,.04,.15,  .3,3*.1,.06,3*.05,.04,
     7 .15,  .301,2*.1,.101,.05,.051,.052,.056,.191,0.,  8*1.,2*0. /,
     2 DSZ /172.,14*2.,160.,15*0.,
     3 172.,3*2.,4.,9*2.,160.,16*0.,  172.,14*2.,150.,15*0.,
     4 170.,27*1.,2.,1.,160.,  170.,27*1.,2.,1.,160.,  170.,25*1.
     5 ,4.,160.,2*0.,  174.,22*1.,164.,  7*0.,  174.,22*1.,164.,  7*0.,
     6 174.,20*1.,2.,4.,160.,  7*0.,  4*1.,10*2.,17*0. /,
     7 IZT /0,169,322,503,898,1292,1633,1944,2254,2538 /
C
      NV=NX(N)
      IL=ILT(N)
      NB=IL+2
      NL=NV+IL
      NU=NL-1
      DO 1 KXR =NB,NU
      IF(X.LT.CURVE(KXR)) GO TO 3
    1 CONTINUE
      KXR = NL
    3 KX= KXR-IL-1
      IL= IZT(N)
      NW= NZ(N)
      NB= IL+2
      NL= NW+IL
      NU=NL-1
      DO 2 KZR=NB,NU
      IF (Z.LT.CURVE(KZR)) GO TO 6
    2 CONTINUE
      KZR =NL
    6 KZ = KZR-IL-1
      J= KZ*NV+ NW+KXR
      DX = (X-CURVE(KXR-1))/DSX(KX,N)
      V= CURVE(J-1)+ DX                       *(CURVE(J)-CURVE(J-1))
      J= J+NV
      VLO = V
      V= CURVE(J-1)+ DX                       *(CURVE(J)-CURVE(J-1))
      V =VLO+(Z-CURVE(KZR-1))/ DSZ(KZ,N)*(V-VLO)
      RETURN
      END
                            CARD COUNT        48
```

ARAP06 FORTRAN Listings (Continued)

```
      SUBROUTINE WING
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /FLAP6/ IDUM1(7),IT8,IT9,IT10,IDUM2(4),NRV,IDUM3(9),NSW,
     1   NSHP,NPDD,IDUM4(3),IT11,IDUM5(10),IT16
      COMMON / WIN /                                   WY(10),WEPE,
     1   PHXYZA(4,5),PHXZT(10,4,3),       WNG(10),HRCP(10),OMSQ( 4),ZM2( 4)
     2   ,CTPC,STPC,EM( 4),PHZV(10, 4),OOA( 4),TPC,SHPE(4,3),
     3               SHPDD(4),TFFY,TFFX,TFFZ,TFMX,TFMY,TFMZ,ETW(3),OFFX(4),
     4   OFFY(4),OFFZ(4),PA(9),PB(9),PC(9),PMTRX(10,4),PHXY(10,4),ALO,
     5   GYMH,HMS,OME,OMES,ALX,ALS,AE,BE,CE,TEX,CLXO
      COMMON /INPU/ DUM6(106),U,ZH,RHO,       DUM8(29),ALW,DUM7(137),SLC
      COMMON /FLAP4/ DUM4(6),PSID,PSIDD,SPSIP,DUM1(7),DT,DUM2(414),DT6,T
      COMMON /WINGT/ XV(15),     VPP(10),VTP(10),PHP(10),ALP(10),CLP(10),
     2   CDP(10),WLP(10),WDP(10),WT(10),WH(10),WO(10),XPDD( 9),YPDD( 9),
     2   ZPDD( 9),OMTRX(10),FFXA,FFZA,FMXA,FMYA,FMZA
      COMMON /FLAP1/ DUM3(174),SCP,SSPP,SPSI,THPDC,STHPP
      COMMON /AIRLD/ DUM5(25),CTHP,STHP,SSPM,CPSIP,DUM9(558),VGM,VGV
      COMMON / FLXOT/ FA1X,FA2X
      DIMENSION                SDD(3, 4),DSDD(3, 4),QC( 4),SHPD( 4),SHP( 4)
     1   ,SHPDDC( 4),SHPJ(3),PHTA(4),PHPA(4),DSHPDD(4),     QB( 4),QA( 4)
      EQUIVALENCE (XDD,XV(1)),(YDD,XV(2)),(ZDD,XV(3)),(T4PD,XV(4)),
     1   (PSIPD,XV(5)),(PHXYZA(1,4),PHTA(1)),                 (THP,XV(9)),
     2   (PSIP,XV(10)),(XDDD,XV(11)),(YDDD,XV(12)),(ZDDD,XV(13)),
     3   (THPDD,XV(14)),(PSIPDD,XV(15)),(PHXYZA(1,5),PHPA(1))
C
      DO 30 J=1,NSHP
      QA(J) = OOA(J)    +TFFX*OFFX(J) + (TFFY-GYMH)*OFFY(J) + TFFZ*
     1   OFFZ(J)     +TFMX*PHPA(J)  +TFMY*PHTA(J)
      QB(J) = -TFFX*OFFZ(J) + TFFZ   *OFFX(J) + TFMZ*PHPA(J)
      QC(J)=TFFY*OFFZ(J) - TFFZ*OFFY(J) - TFMZ*PHTA(J)
   30 CONTINUE
      DO 90 NT=1,10
      PHTASP = 0.D0
      PHPASP = 0.D0
      DO 125 J=1,NSHP
      SHPD(J) = SHPE(J,1)+ (SHPDD(J)+SHPE(J,3))*DT
      SHP(J)  = SHPE(J,2)+ (SHPD(J) +SHPE(J,1))*DT + (SHPDD(J) -
     1   SHPE(J,3))*DT6
      PHTASP = PHTASP + PHTA(J)*SHP(J)
      PHPASP = PHPASP + PHPA(J)*SHP(J)
  125 CONTINUE
      IF (RHO.EQ.0.D0 .OR. U.EQ.0.D0) GO TO 26
      DO 50 I=1,NSW
      VP = VGV
      VT = U+VGM
      AL = ALW
      DO 40 J=1,NSHP
      VP = VP - PHZV(I,J)*SHPD(J)
      VT = VT + PHXY(I,J)*SHPD(J)
   40 AL = AL + PHXZT(I,J,3)*SHP(J)
      PH = DATAN2(VP,VT)
      ALO = DCOS(AL)
      AL1 = DSIN(AL)
      AL = AL + PH
      VSQ = VT*VT + VP*VP
```

ARAP06 FORTRAN Listings (Continued)

```
      V =DSQRT(VSQ)
      CL = SLC*AL
      CD = .0087 - .0216*AL + .4*AL*AL
      WL = CL*VSQ*HRCR(I)
      WD = CD*VSQ*HRCR(I)
      WT(I) = (WL*VT + WD*VP)/V
      WH(I) = (WD*VT - WL*VP)/V
      WQ(I) = ((ALO*VT - ALI*VP)*WL + (ALI*VT + ALO*VP)*WD)*WEPE/V
      IF (NPV.NE.2 .AND. PSID.NE.0.D0) GO TO 50
      VPP(I) = VP
      VTP(I) = VT
      PHP(I) = 57.3*PH
      ALP(I) = 57.3*AL
      CLP(I) = CL
      CDP(I) = CD
      WLP(I) = WL
      WDP(I) = WD
50    CONTINUE
26    JC = 0
      DO 80 J=1,NSHP
      WARLD = 0.D0
      DO 70 I=1,NSW
70    WARLD = WARLD - WH(I)*PHXZT(I,J,1) + WQ(I)*PHXZT(I,J,3) + WT(I)*
     1 PHXZT(I,J,2) - (PHXZT(I,J,2) + WT(I)*PHXZT(I,J,3)*ALO)*WMG(I)
      Q     = WARLD + QA(J) + QB(J)*PHTASP + QC(J)*PHPASP
      SHPDDC(J) = Q    /EM(J) - OMSQ(J)*SHP(J) - ZM2(J)*SHPD(J)
      DSHPDD(J) = SHPDDC(J) - SHPDD(J)
      IF(DABS(DSHPDD(J)).GT. DMAX1(5.D-4,.005*DABS(SHPDDC(J)))) JC=JC+1
80    CONTINUE
      IF (JC.EQ.0                ) GO TO 165
      DO 99 J=1,NSHP
      SDD(1,J) = SHPDD (J)
      DSDD(1,J) = DSHPDD(J)
      CALL PSLOP (SDD(1,J),DSDD(1,J),WT)
      SHPDD(J) = SDD(1,J)
99    CONTINUE
165   DO 22 L=1,15
22    XT(L) = 0.D0
      THP    = TPC
      ETW(1) = 0.D0
      ETW(2) = 0.D0
      ETW(3) = 0.D0
      DO 170 J=1,NSHP
      IF ((T)4.EQ.0) GO TO 34
      SHPJ(3)    =(SHPDDC(J) + SHPE(J,3))/2.D0
      SHPJ(1)  =(SHPD(J) + SHPE(J,1))/2.D0
      SHPJ(2)    =(SHP(J) + SHPE(J,2))/2.D0
      GO TO 36
34    SHPJ(3)    = SHPDDC(J)
      SHPJ(1)  = SHPD(J)
      SHPJ(2)    = SHP(J)
36    DO 170 K=1,3
      SHPE(J,K) = SHPJ(K)
      DO 32 M=1,5
      L = (K-1)*5 + M
```

ARAP06 FORTRAN Listings (Continued)

BELL HELICOPTER COMPANY

```
32    XY(L) = PHXYZA(J,M)*SHPJ(K) + XY(L)
      ETH(K) = PHXZT(NSH,J,K)*SHPJ(2) + ETH(K)
170   CONTINUE
      IF (IT11.EQ.0) GO TO 24
      DO 20 I=1,10
      OPTRX(I) = 0.D0
      DO 20 J=1,NSHP
20    OPTRX(I) = OMTRX(I) + RMTRX(I,J)*SHPE(J,2)
24    STHP  =DSIN(THP)
      CTHP  =DCOS(THP)
      SPSIP =DSIN(PSIP)
      CPSIP =DCOS(PSIP)
      THPDC = THPD*CPSIP
      STHPP = STHP*SPSIP
      SCP  = STHP*CPSIP
      CCP = CPSIP*CTHP
      TSCP  = THPD*SCP + PSIPD*CTHP*SPSIP
      TCCP  = THPDC*CTHP - PSIPD*STHPP
      FA1X = XODD*CCP - XOD *TSCP - YODD*SPSIP - YOD*PSIPD*CPSIP +
     1  ZOD*TCCP + ZODD*SCP
      FA2X  = XOD*CCP - YOD*SPSIP + ZOD*SCP
      CTHPP = CTHP*SPSIP
      IF (ITR.NE.2) GO TO 28
      AED = XOD - (PSIPD*CTHPP + THPD*SCP)*CE - AE*THPD*CTHP + (PSIPD*
     1  CCP - THPD*STHPP)*BE + U
      CED =-ZOD + (PSIPD*STHPP - THPD*CCP)*CE + AE*THPD*STHP - (PSIPD*
     1  SCP + THPD*CTHPP)*BE
      VEF = DSQRT(AED*AED + CED*CED)*HRS
      AEF = DATAN2(CED,AED) + THP + ALX
      IF (T.GT.TEX) AEF = AEF + ALS*DSIN((T-TEX)*OMES+OME)*(T-TEX))
      CLX = CLXO*AEF
      CDX = 8.7D-3 - 2.16D-2*AEF + .4D0*AEF*AEF
      SA = STHP*AED + CTHP*CED
      CA = CTHP*AED - STHP*CED
      FFXA = (SA*CDX + CA*CLX)*VEF
      FFZA = (SA*CLX - CA*CDX)*VEF
      FMXA = FFZA*BE
      FMYA = (CE-ZM)*FFXA - FFZA*AE
      FMZA = -FFXA*BE
28    IF (IT10.EQ.0) RETURN
      PSPDSQ = PSIPD*PSIPD
      PDTDP = PSIPD*THPD
      THPDSQ = THPD*THPD
      XPDDA = THPDD*CTHP - THPDSQ*STHP
      XPDDB = PSIPDD*CCP - PSPDSQ*CTHPP - 2.*PDTDP*SCP - THPDD*STHPP -
     1        THPDSQ*CTHPP
      XPDDC = PSIPDD*CTHPP + PSPDSQ*CCP - 2.*PDTDP*STHPP + THPDD*SCP +
     1        THPDSQ*CCP
      YPDDB = PSIPDD*SPSIP + PSPDSQ*CPSIP
      YPDDC = PSIPDD*CPSIP - PSPDSQ*SPSIP
      ZPDDA = THPDD*STHP + THPDSQ*CTHP
      ZPDDB = PSIPDD*SCP - PSPDSQ*STHPP + 2.*PDTDP*CCP + THPDD*CTHPP -
     1        THPDSQ*STHPP
      ZPDDC = PSIPDD*STHPP + PSPDSQ*SCP + 2.*PDTDP*CTHPP - THPDD*CCP +
     1        THPDSQ*SCP
```

ARAP06 FORTRAN Listings (Continued)

PAGE 6

```
      DO 35 K=1,NPDD
      XDD     = (XODD - PC(K)*XPDDC + PB(K)*XPDDB - PA(K)*XPDDA)/ 386.
      YDD     = (YODD - PC(K)*YPDDC - PB(K)*YPDDB) / 386.
      ZDD     = (ZODD - PC(K)*ZPDDC + PB(K)*ZPDDB - PA(K)*ZPDDA) / 386.
      XPDD(K) = ZDD*CTHP - XDD*STHP
      YPDD(K) = XDD*CTHPP + YDD*CPSIP + ZDD*STHPP
      ZPDD(K) = XDD*CCP - YDD*SPSIP + ZDD*SCP
   35 CONTINUE
      RETURN
      END
                        CARD COUNT      175
```

ARAP06 FORTRAN Listings (Concluded)

BELL HELICOPTER COMPANY

```
      COMMON/CONTRL/CON ,NAERO,NSIZE,MODES,BREF,NVEL,NSIZ,ICT,VELCTY,RHO
     1,FK,NAER
      COMMON/MCO4/AM,MSB
      COMMON/MATRS/F,DUM,A,C,B,DD,TTL,    X,Y,M,SK,EK
      COMPLEX *16 F(74,74),DUM(74)
      INTEGER *2 MSB(150)
      REAL *8 A(74 ,74), CON,AM(150),TTL(10),X(22),Y(22),M(22),SK(22)
     1,EK(22) ,C(2,2),B(2,2),D(74,74),DD(222)
      DIMENSION VELCTY(15),FMT(29)
      EQUIVALENCE(D(1,1),F(1,1))
      DATA FMT/'FOLD-UNFOLD ROTOR VIBRATION AND FLUTTER ANALYSIS BY A CO
     1LLOCATION METHOD USING AERODYNAMIC INFLUENCE COEFFICIENTS    '/
      CALL ERRSET(208,0,-1,0,0,0)
 991  READ (5, 1100,END=8) TTL
 1100 FORMAT ( 10A8)
      WRITE(6,6) FMT, TTL
 6    FORMAT('1',4X,29A4 /'0',10A8)
      READ          (5       , 1) NSIZ ,NAER ,MODES,FK,AMK,RHO
 1    FORMAT(3I4,3F12.5)
      NAERO=IABS(NAER)
      NSIZE=IABS(NSIZ)
      IF(AMK.EQ.0.)  AMK = 1.
      IF(RHO.EQ.0.) RHO=.002378
      IF(FK.EQ.0.) FK=1.
      IF(MODES.EQ.0) MODES=NSIZE
      WRITE(6,3) NSIZ ,NAER, MODES,FK, AMK, RHO
 3    FORMAT(/'CONTROL DATA'/'NSIZE',I15/' NAERO',I14/' MODES OUT',I12
     1/' FLEX K',E15.5/' MASS K',E15.5/' DENSITY',E14.5)
      AMK=AMK/32.174
      DO 5 I=1,15
 5    VELCTY(I)=0.
      IF (NAERO.EQ.0 ) GO TO 9
      REWIND 1
      READ(1,2) BR,S,BREF,              (VELCTY(I),I=1,NAERO)
    2 FORMAT (6D12.8)
      WRITE(6,4) BR,S,BREF,(VELCTY(I),I=1,NAERO)
 4    FORMAT( ' BR',F19.5/' S',E20.5/' B(REF)',E15.5/'0REDUCED VELOCITIE
     1S'//(F20.5,5F15.5))
      CON = RHO        *BR**2*S
      CALL DFAE
 9    CALL STGN(NSIZE)
      FK=FK/12.
      WRITE(6,14)
   14 FORMAT (1H1 43X, 19H FLEXIBILITY MATRIX / )
      CALL MPRINT (A,NSIZE,NSIZE,74)
      DO 17 J=1,NSIZE
      DO 12 I=1,NSIZE
      D(I,J)=0.D0
 12   A(I,J)=A(I,J)*FK
 17   IF(NAERO.GT.0) WRITE(1)(A(I,J),I=1,NSIZE)
      ICT=0
      NV=-2
      DO 7 I=1,22
      C(2,2) =  1.D0/(K(I)*Y(I))
      C(1,2) =X(I)*C(2,2)
```

NOT REPRODUCIBLE

DFAL17 FORTRAN Listings

```
      C(1,1)= Y(I)*C(2,2)
      C(2,1) =  -C(2,2)
      B(1,1) =  M(I)
      B(1,2) =  SX(I)
      B(2,1) =  SX(I)
      B(2,2) =  IX(I)
      IF(I.LT.4) GO TO 11
      DO 13 K=1,2
      NV=NV+1
      IC=I+I+NV
      ICT=ICT+1
      AM(ICT)=M(I)*AMK
      MSH(ICT)=IC* NSIZE+IC-NSIZE
   13 D(IC,IC)=M(I)
   11 IC=I+I+NV
      CALL DPROD(C,B,2,TTL)
      DO 7 K=1,2
      ICK=K+IC
      DO 7 J=1,2
      IF(B(J,K).EQ.0.D0) GO TO 7
      ICT=ICT+1
      IF(ICT.GT.150) GO TO 15
      ICJ=J+IC
      AM(ICT)= B(J,K)*AMK
      MSH(ICT)= NSIZE*ICK+ICJ-NSIZE
      D(ICJ,ICK)=B(J,K)
    7 CONTINUE
      WRITE(6,10)
   10 FORMAT  (1H1 ,49X, 12H MASS MATRIX /  )
        CALL MPRINT (D,NSIZE,NSIZE,74)
      CALL PRT3
      GO TO 991
   15 WRITE(6,16)
   16 FORMAT('0TOO MANY NON-ZERO MASSES')
    8 STOP
      END                                                        DF041010
                           CARD COUNT       91
```

DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE ALLMAT (M,NCAL)
      COMMON /MATRS/A,LAMBDA,V,VECT,H,MULT,SHIFT,ING,NS,INTH
      COMPLEX *16 A(74,74), H (2921 ),LAMBDA(74),VECT(74),MULT(74),
     1 SHIFT(3),TEMP,SIN,COS,DMA,DMC,COG,SIG,V(149)
      LOGICAL*1 INTH(74),TWICE
      INTEGER*2 ING(74),NS(74)
      INTEGER R,RP1,RP2
      REAL *8 BIG, SUM, EPS,CDAB
      CDAB(SIG)=SIG*DCONJG(SIG)
      N=M                                                               ALLM0130
      SHIFT(1)=(0.D0,0.D0)
C     REDUCE MATRIX A TO HESSENBERG FORM                                ALLM0330
      NM2=N-2
      DO 15 R=1,NM2                                                     ALLM0360
      RP1=R+1                                                           ALLM0370
      RP2=R+2                                                           ALLM0380
      BIG =0.D0
      INTER=RP1
      DO 5 I=RP1, N
      SUM =CDAB(A(I,R))
      IF(SUM.LE.BIG) GO TO 5
      INTER=I
      BIG = SUM
    5 CONTINUE                                                          ALLM0460
      ING(R)=INTER
      IF(BIG.EQ.0.D0) GO TO 15
      IF(INTER.EQ.RP1)GO TO 8                                           ALLM0490
      DO 6 I=R,N                                                        ALLM0500
      TEMP=A(RP1,I)                                                     ALLM0510
      A(RP1,I)=A(INTER,I)                                               ALLM0520
    6 A(INTER,I)=TEMP                                                   ALLM0530
      DO 7 I=1,N                                                        ALLM0540
      TEMP=A(I,RP1)                                                     ALLM0550
      A(I,RP1)=A(I,INTER)                                               ALLM0560
    7 A(I,INTER)=TEMP                                                   ALLM0570
8     TEMP = A(RP1,R)
      DO 9 I=RP2,N
      MULT(I) = A(I,R) /TEMP
9     A(I,R) =MULT(I)
      DO 13 I=1,N
      TEMP =A(I,RP1)
      IF(I.GT.RP1) TEMP=TEMP- MULT(I)*A(RP1,RP1)
      DO 12 J=RP2,N                                                     ALLM0680
   12 TEMP=TEMP+A(I,J)*MULT(J)                                          ALLM0690
13    A(I,RP1) = TEMP
      DO 14 I=RP2,N                                                     ALLM0710
      DO 14 J=RP2,N                                                     ALLM0720
   14 A(I,J)=A(I,J)-MULT(I)*A(RP1,J)                                    ALLM0730
   15 CONTINUE                                                          ALLM0740
C     CALCULATE EPSILON                                                 ALLM0760
      EPS= 0.D0
      DO 16 I=1,N                                                       ALLM0790
   16 EPS=EPS+CDABS(A(I,I))                                             ALLM0800
      DO 19 I=2,N                                                       ALLM0810
      SUM =0.D0
```

DFAL17 FORTRAN Listings (Continued)

```
      IM1=I-1                                                           ALLM0830
      DO 17 J=IM1,N                                                     ALLM0840
   17 SUM=SUM+CDABS(A(I,J))                                             ALLM0850
   18 EPS = DMAX1(EPS,SUM)
      EPS= DSQRT(   DFLOAT(N)) *EPS*1.D-18
      IF(EPS.EQ.0.D0) EPS =1.D-18
      K=0
      DO 19 J=1,N
      JP = J+2
      DO 19 I=1,JP
      K=K+1
      IF(I.EQ.J) NS(J)=K
      IF(I.GT.N) GO TO 19
      H(K) = A(I,J)
   19 CONTINUE
   20 IF(N.NE.1)GO TO 21                                                ALLM0920
      LAMBDA(M)=H(1   )+SHIFT(1)
      GO TO 3
   24 LAMBDA(MN1)=H(NL  )+SHIFT(1)
      N=N-1                                                             ALLM1040
   21 ICOUNT=0                                                          ALLM1030
      IF(N.EQ.2)GO TO 2
      NCAL=M-N                                                          ALLM1360
      NL=NS(N)
      NMS=N-1
      NMA =NS(NMS)
      MN1=M-NMS
      NM1 =NMA+1
      NM2 =NS(N-2)+1
   22 IF(CDABS (H(NL  )).EQ.0.D0) GO TO 23
      TEMP =H(NM1)/H(NL)
      IF(CDABS (TEMP) .LE. 1.D-20) GO TO 24
   23 IF(CDABS(H(  NM1)).LT. EPS ) GO TO 24
C     DETERMINE SHIFT                                                   ALLM1080
      DMA =H(NMA)+ H(NL)
      DMC =H(NL)*H(NMA)-H(NM1)*H(NL-1)
      SHIFT(2) =(DMA +CDSQRT(DMA**2-(4.D0,0.D0)*DMC))*(.5D0,0.D0)
      SHIFT(3) = DMA
      IF(CDABS (SHIFT(2)).NE.0.D0) SHIFT(3) =DMC /SHIFT(2)
      INDEX=3                                                           ALLM1190
      IF(CDABS (SHIFT(2)-H(NL  )).LT. CDABS (SHIFT(3)-H(NL  ))) INDEX =2
      IF(CDABS (H(NM1)).GE.EPS**2)GO TO 30
      LAMBDA(MN1)=SHIFT(2)+SHIFT(1)                                     ALLM1220
      LAMBDA(MN1+1)=SHIFT(3)+SHIFT(1)                                   ALLM1230
      N=N-2                                                             ALLM1250
      GO TO 20                                                          ALLM1270
   30 SHIFT(1)=SHIFT(1)+SHIFT(INDEX)                                    ALLM1280
      DO 31 I=1,N                                                       ALLM1290
      NMC= NS(I)
   31 H(NMC) =H(NMC)- SHIFT(INDEX)
C     PERFORM GIVENS ROTATIONS, QR ITERATES                             ALLM1320
      IF(ICOUNT.GT.10)GO TO 1
      ICOUNT=ICOUNT+1
      DMA =H(1)
      DMC =H(2)
```

DFAL17 FORTRAN Listings (Continued)

```
      DO 36 R=1,NMS
      SUM =CDABS(DMA)+CDABS(DMC)
      IF(SUM.EQ. 0.D0) GO TO 22
      SIG=DSQRT(SUM)
      RP1=R+1                                                ALLM1420
      RP2 =R+2
      COS = DMA /DCMPLX(SIG,0.D0)
      SIN = DMC /DCMPLX(SIG,0.D0)
      COG=DCONJG(COS)
      SIG=DCONJG(SIN)
      INDEX=MAX0(R-1,1)                                      ALLM1480
      NMC = NS(INDEX)-INDEX+R
      DO 33 I=INDEX,N                                        ALLM1490
      NMD = NMC+1
      TEMP= COG*H(NMC)+SIG*H(NMD)
      H(NMD)=-SIN*H(NMC)+ COS*H(NMD)
      H(NMC)=TEMP
   33 NMC = NMC+I+2
      NM= NS(M)-R
      DO 34 I=1,R
      NMC =NM+I
      NMD =NMC+RP2
      TEMP =COS*H(NMC)+SIN*H(NMD)
      H(NMD) =-SIG*H(NMC)+COG* H(NMD)
   34 H(NMC) = TEMP
      INDEX = MIN0 (RP2,N)
      DMA = H(NS(RP1))
      DMC = H(NS(RP1)-RP1+INDEX)
      DO 35 I=RP1,INDEX                                      ALLM1600
      NMC=NM +I
      NMD=NMC+RP2
      H(NMC)= SIN*H(NMD)
   35 H(NMD) =COG*H(NMD)
   36 CONTINUE                                               ALLM1630
      GO TO 22                                               ALLM1650
    2 DMA = H(1)+H(5)
      DMC = H(5)*H(1)-H(2)*H(4)
      TEMP = (DMA+ CDSQRT( DMA**2 -(4.D0,0.D0)*DMC))*(.5D0,0.D0)
      LAMBDA(M) = SHIFT(1)+TEMP
      LAMBDA(M-1) = DMA + SHIFT(1)
      IF(CDABS (TEMP).NE.0.D0 ) LAMBDA(M-1) = DMC/TEMP +SHIFT(1)
    3 NCAL=M
    1 N=M
      NM1=N-1                                                ALLM1710
      NM2=N-2                                                ALLM2260
      NL = NS(N)
      GO TO 37                                               ALLM0310
      ENTRY ALLVEC(L)
C     CALCULATE VECTORS                                      ALLM1670
      K=0
      DO 75 J= 1,N
      JP = J+2
      DO 40 I=1,JP
      K=K+1
      IF(I.GT.N) GO TO 40
```

DFAL17 FORTRAN Listings (Continued)

```
      H(K) = A(I,J)
40    CONTINUE
39    HENS(JJ)=HENS(JJ)-LAMBDA(L)
      DO 44 I=1,NM1                                              ALLM1840
      MULT(I) = (0.D0,0.D0)
      INTH(I)=.FALSE.                                            ALLM1860
      IP1=I+1                                                    ALLM1870
      NMA =NS(I)
      NMB = NMA+1
      IF(CDABS(H(NMB)).LE. CDABS(H(NMA))) GO TO 42
      INTH(I)=.TRUE.                                             ALLM1890
      NMC = NMA
      DO 41 J=I,N                                                ALLM1900
      NMD = NMC+1
      TEMP=H(NMD)
      H(NMD) =H(NMC)
      H(NMC)= TEMP
41    NMC = NMC+J+2
42    IF(CDABS(H(NMA)).EQ.0.D0) GO TO 44
      MULT(I)=-H(NMB)/H(NMA)
      NMC= NS(IP1)
      DO 43 J=IP1,N                                              ALLM1960
      NMD= NMC-1
      H(NMC) = H(NMC)+MULT(I)* H(NMD)
43    NMC= NMC+J+2
  44  CONTINUE                                                   ALLM1980
      DO 45 I=1,N                                                ALLM1990
45    VECT(I) =(1.D0,0.D0)
      TWICE=.TRUE.                                               ALLM2230
      GO TO 46                                                   ALLM2240
52    DO 51 I=1,NM1                                              ALLM2170
      IP1= I+1
      IF(.NOT.INTH(I))GO TO 51                                   ALLM2180
      TEMP=VECT(I)                                               ALLM2190
      VECT(I) =VECT(IP1)
      VECT(IP1) = TEMP
51    VECT(IP1) =VECT(IP1)+ MULT(I)*VECT(I)
      TWICE=.FALSE.                                              ALLM2010
46    IF(CDABS(H(NL)).EQ.0.D0) H(NL) =DCMPLX(EPS,0.D0)
      VECT(N) =VECT(N)/H(NL)
      DO 48 I=1,NM1                                              ALLM2040
      K=N-I                                                      ALLM2050
      KP1 =K+1
      NMC =NS(K)
      NMD =NS(KP1)-1
      DO 47 J=KP1,N
      VECT(K)=VECT(K)-H (NMD)*VECT(J)
47    NMD =NMD+J+2
      IF(CDABS (H (NMC)).EQ.0.D0) H (NMC) =DCMPLX(EPS,0.D0)
  48  VECT(K)=VECT(K)/H (NMC)
      BIG =CDABS(VECT(1))
      DO 49 I=2,N
49    BIG = DMAX1(BIG,CDABS(VECT(I)))
      DO 50 I=1,N                                                ALLM2140
50    VECT(I) = VECT(I) /DCMPLX(BIG,0.D0)
```

DFAL17 FORTRAN Listings (Continued)

```
      IF(IWICC)GO TO 52                                               ALLM2160
      DO 56 I=1,NM2
      NII =NM1-I
      NI1 =NII+2
      DO 53 J=NI1,N                                                   ALLM2300
   53 VECT(J)=A(J,NII)*VECT(NII+1)+VECT(J)
      INDEX=ING(NII)                                                  ALLM2320
      TEMP=VECT(NII+1)                                                ALLM2330
      VECT(NII+1)=VECT(INDEX)                                         ALLM2340
   56 VECT(INDEX)=TEMP                                                ALLM2350
   37 RETURN
      END
                        CARD COUNT       232
```

DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE BESJ(X,N,BJ,D,IER)                                     DF040380
      BJ=0.                                                             DF040390
      IER=1
      IF(N)200,20,20
   20 IER=2
      IF(X)200,200,31
   31 IF(X-15.) 32,32,34                                                DF040460
   32 NTEST= 20.+10.*X-X**2/3.                                          DF040470
      GO TO 36                                                          DF040480
   34 NTEST= 90.+X/2.                                                   DF040490
   36 IF(N-NTEST) 40,38,38                                              DF040500
   38 IER=4                                                             DF040510
      GO TO 200
   40 IER=0                                                             DF040530
      BPREV=0.                                                          DF040550
      IF(X-5.) 50,60,60                                                 DF040590
   50 MA=X+6.                                                           DF040600
      GO TO 70                                                          DF040610
   60 MA=1.4*X+60./X                                                    DF040620
   70 MZERO=MAX0(MA,N+IFIX(X)/4+2)
      MMAX=NTEST                                                        DF040650
      DO 190 M=MZERO,MMAX,3
      FM1=1.0E-28                                                       DF040730
      FM=0.                                                             DF040740
      ALPHA=0.                                                          DF040750
      IF(MOD(M,2))120,110,120
  110 JT=-1                                                             DF040770
      GO TO 130                                                         DF040780
  120 JT=1                                                              DF040790
  130 M2=M-2                                                            DF040800
      DO 160 K=1,M2                                                     DF040810
      MK=M-K                                                            DF040820
      BMK=2.*FLOAT(MK)*FM1/X-FM                                         DF040830
      FM=FM1                                                            DF040840
      FM1=BMK                                                           DF040850
      IF(MK.EQ.N+1)BJ=BMK
      JT=-JT
  160 ALPHA=ALPHA+BMK*FLOAT(JT+1)
      BMK=2.*FM1/X-FM                                                   DF040910
      IF(N.EQ.0)BJ=BMK
      ALPHA=ALPHA+BMK
      BJ=BJ/ALPHA                                                       DF040950
      IF(ABS(BJ-BPREV)-ABS(D*BJ)) 200,200,190                           DF040960
  190 BPREV=BJ                                                          DF040970
      IER=3                                                             DF040980
  200 RETURN                                                            DF040990
      END                                                               DF041000
                              CARD COUNT          47
```

NOT REPRODUCIBLE

DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE BESY(X, BY,IER)
      REAL*8 C
      REAL BY(2)
      IER=2
      IF(X)170,170,20
   20 IER=0
      IF(X-4.0)40,40,30
   30 T1=4.0/X                                                  BESV 550
      T2=T1*T1                                                  BESV 560
      P0=((((-.0000037043*T2+.0000173565)*T2-.0000487613)*T2   BESV 570
     1  +.00017343)*T2-.0017530621)*T2+.3989423                 BESV 580
      Q0=((((.0000032312*T2-.0000142078)*T2+.0000342468)*T2    BESV 590
     1  -.0000869791)*T2+.0004564324)*T2-.01246694              BESV 600
      P1=((((.0000042414*T2-.0000200920)*T2+.0000580759)*T2    BESV 610
     1  -.0002232031)*T2+.0029218256)*T2+.3989423               BESV 620
      Q1=((((-.0000036594*T2+.0000162200)*T2-.0000398708)*T2   BESV 630
     1  +.0001064741)*T2-.0006390400)*T2+.03740084              BESV 640
      A=2.0/DSQRT(DBLE(X))
      B=A*T1                                                    BESV 650
      C=X-.7853982                                              BESV 670
      SC=DSIN(C)
      CC=DCOS(C)
      Y0=A*P0*SC    +B*Q0*CC
      Y1=-A*P1*CC    +B*Q1*SC
      GO TO 90                                                  BESV 730
   40 XX=X/2.                                                   BESV 740
      XT=XX*XX
      T=ALOG(XX)+.5772157                                       BESV 750
      SUM=0.                                                    BESV 770
      TERM=XT*(1.-T)
      Y0=T+TERM
      DO 70 L=2,16
      FL=L
      SUM=SUM+1./FLOAT(L-1)
      TS=T-SUM                                                  BESV 940
      TERM= TERM*(XX/FL)**2*(1./(FL*TS)-1.)
   70 Y0=Y0+TERM
      TERM = XX*(T-.5)
      SUM=0.                                                    BESV 880
      Y1=TERM
      DO 80 L=2,16
      FL1=L-1
      SUM=SUM+1./FL1
      FL=L                                                      BESV 920
      TS=T-SUM
      TERM=TERM*XT    /(FL1*FL    )*((.5/FL-TS)/(TS+.5/FL1))
   80 Y1=Y1+TERM
      PI2=.6366198                                              BESV 970
      Y0=PI2*Y0                                                 BESV 980
      Y1=PI2*(Y1-1./X)
   90 BY(1)=Y0
      BY(2)=Y1
  170 RETURN                                                    BESV1290
      END                                                       BESV1340
                              CARD COUNT        54
```


DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE DFAE
      COMMON/MATRS/ A,CHI,PARTA,AM,BM,FJDK,EVOK
      COMMON/CONTRL/CON ,JSZ  ,NSIZE,MODES,MREF,J    ,NSIZ,ICT,FKR(15)
      REAL    COSLMA(4),BRA(4),SA(4),CAPA(4),CBAA(4)
     1,AM(2,2  ),BM(2,4),B(7,4),EVOK(2),FJDK(2),DELTAV(7,4),D(7,4)
      REAL*8 CHI(2,4),PARTA(2,4),CON
      COMPLEX*16 A(74,74)
      INTEGER*2  ISZR(4)
      DATA N/2/,N2/4/,PI/3.141593/,NPRO/4/,ISZR/7,5,5,5/
      WRITE(6,4)
    4 FORMAT(//'AERODYNAMIC INPUT DATA' /// '  SURFACE',2X,'COSINE LAMB
     1DA',17X,'BR',15X,'S',14X,'CAP S',12X,'C BAR' // )
      DO 2  JKL=1,NPRO
      READ(5,3) COSLMA(JKL),BRA(JKL),SA(JKL),CAPA(JKL),CBAA(JKL)
    3 FORMAT (6F12.4)
      ISZ=ISZR(JKL)
      WRITE(6,1) JKL, COSLMA(JKL),BRA(JKL),SA(JKL),CAPA(JKL),CBAA(JKL)
    1 FORMAT (I6,5E17.5)
      READ(5,3)(DELTAV(I,JKL),I=1,ISZ)
      READ(5,3) (B(I,JKL),I=1,ISZ)
    2 READ(5,3) (D(I,JKL),I=1,ISZ)
      WRITE (6,9)
    9 FORMAT(//5X,'STRIP',3X,'1',15X,'2',15X,'3',16X,'4',16X,'5',16X,'6'
     1,16X,'7'/)
      DO 10JKL=1,NPRO
      IF(ISZR(JKL)-6)6,10,11
    6 WRITE(6,5) (DELTAV(I,JKL),I=1,5),JKL,(B(I,JKL),I=1,5),
     1 (D(I,JKL),I=1,5)
    5 FORMAT('0SURFACE  DV',5E17.5/I5,6X,'B',5F17.5/11X,'D',5E17.5)
      GO TO 10
   11 WRITE(6,7) (DELTAV(I,JKL),I=1,7),JKL,(B(I,JKL),I=1,7),
     1 (D(I,JKL),I=1,7)
    7 FORMAT('0SURFACE  DV',7E17.5/I5,6X,'B',7F17.5/11X,'D',7E17.5)
   10 CONTINUE
      RETURN
      ENTRY DF67
      IST=-1
      IV=2
      DO 40 JKL=1,NPRO
      ISZ=ISZR(JKL)
      SCACON=PI/SA(JKL)*COSLMA(JKL)/BRA(JKL)**2*CON
      DO 60 I=1,ISZ
      IF(IST.EQ.13) IV=4
      IST=IST+IV
      SCAFF    =SCACON*B(I,JKL)**2*DELTAV(I,JKL)
      RID=B(I,JKL)/D(I,JKL)
      AM(1,1  ) =1.
      AM(1,2  ) =-RID
      AM(2,1  ) =0.
      AM(2,2  ) = RID
      FIK=FK-(J)*BRA(JKL)/B(I,JKL)
      IF(FIK.NE.0.) GO TO 36
      F=.5
      G=0.
      GO TO 35
```

DFAL17 FORTRAN Listings (Continued)

```
   34 EKNOW =1. /EIK
      CALL BESY (EKNOW,      EYOX    , IEY)
      DO 8 K=1,2
      CALL BESJ (EKNOW,K-1, FJOX(K),1.E-5,IEJ)
    8 IF((IEJ+IEY.NE.0) WRITE(6,12) JKL,J,I,K, EKNOW
   12 FORMAT (*0ERROR IN BESSEL FUNCTION,SURFACE*,I3,*  VELOCITY*,I3,*
     1STRIP*,I3,*  ORDER*, I3,*  K VALUE*, F12.5)
      DEN=(EJOX(2)+EYOX(1))**2+(EYOX(2)-FJOX(1))**2
      F=(EJOX(2)*(EJOX(2)+EYOX(1))+EYOX(2)*(EYOX(2)-EJOX(1)))/DEN
      G= -(EYOX(2)*EYOX(1)+EJOX(2)*EJOX(1))/DEN
   35 EK2 =EIK*EIK
      TSA=2.*G*EIK
      TSB=2.*F*EIK
      BM(1,1)=1.+TSA
      BM(1,2)=-TSB
      BM(1,3) = .5*TSA-2.*F*EK2
      BM(1,4) = -EIK-TSB- 2.*G*EK2
      BM(2,1) =.5
      BM(2,2) =0.
      BM(2,3) =.375
      BM(2,4) =-EIK
      DO 82 K=1,N
      DO 82 L=1,N2
      PART A(K,L) =0.D0
      DO 82JJ=1,N
   82 PART A(K,L) = PART A(K,L)+AM(K,JJ  )* BM(JJ,L)
      DO 81 L=1,N2
      M=(L+1)/2
      LM=MOD(L,2)
      DO 81 K=1,N
      CHI (K,L  )=0.D0
      DO 83JJ=1,N
   83 CHI (K,L  ) =CHI(K,L  ) +PART A  (K,2*JJ-LM)*AM(M,JJ)
   81 CHI(K,L)=CHI(K,L)*SCALER
      ISA=IST-1
      DO 60 K=1,N
      ISK=ISA+K
      M=0
      DO 60 L=1,N2,2
      M=M+1
   60 A(ISA+M,ISK)=DCMPLX(CHI(K,L),CHI(K,L+1))
      RETURN
       END                                                     HM079203
                               CARD COUNT       98
```

DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE OPROD(A,B,N,S)
      REAL *8 A(N,N), B(N,N), S(N),SAVE
      NM1=N-1
      DO 100 I=1,2
      IF(I.EQ.1) GO TO 1
      DO 10 II=1,NM1
      III = II + 1
      DO 10 IK=III,N
      SAVE = B(II,IK)
      B(II,IK) = B(IK,II)
   10 B(IK,II) = SAVE
    1 DO 100 J=1,N
      DO 50 K=1,N
      S(K) = 0.D0
      DO 50 L=1,N
   50 S(K)= S(K) + B(J,L)*A(L,K)
      DO 100 KJ=1,N
  100 B(J,KJ) = S(KJ)
      RETURN
      END
                              CARD COUNT        20
```

DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE INVERS(N)
      COMMON/MATRS/DM(11102),STIFF(74,74),NDEX
      REAL*8 BIGA,HOLD,STIFF,DM
      INTEGER   NDEX(74)
      DATA  NSZ/74/
      IROW=1
      JCOL=1
C     SEARCH FOR LARGEST ELEMENT                                STIFFINV
      DO 80 K=1,N                                               STIFFINV
      IF(K.EQ.N) GO TO 45                                       STIFFINV
      BIGA=0.DO                                                 STIFFINV
      DO 20 J=K,N                                               STIFFINV
      DO 20 I=K,N                                               STIFFINV
      HOLD = DABS(STIFF(I,J))
      IF(BIGA.GE.HOLD) GO TO 20                                 STIFFINV
      BIGA = HOLD                                               STIFFINV
      IROW =I                                                   STIFFINV
      JCOL =J                                                   STIFFINV
   20 CONTINUE                                                  STIFFINV
C     INTERCHANGE ROWS                                          STIFFINV
      NDEX(K) = JCOL * NSZ - NSZ + IROW                         STIFFINV
      IF(IROW.EQ.K) GO TO 35
      DO 30 I=1,N                                               STIFFINV
      HOLD =-STIFF(K,I)                                         STIFFINV
      STIFF(K,I) = STIFF(IROW,I)                                STIFFINV
   30 STIFF(IROW,I) = HOLD                                      STIFFINV
C     INTERCHANGE COLUMNS                                       STIFFINV
   35 IF(JCOL.EQ.K) GO TO 45
      DO 40 J=1,N                                               STIFFINV
      HOLD = -STIFF(J,K)                                        STIFFINV
      STIFF(J,K) = STIFF(J,JCOL)                                STIFFINV
   40 STIFF(J,JCOL) = HOLD                                      STIFFINV
C     DIVIDE COLUMN BY MINUS PIVOT                              STIFFINV
   45 IF(STIFF(K,K).EQ.0.DO)GOTO 25
      BIGA =-STIFF(K,K)                                         STIFFINV
      DO 55 IC=1,N                                              STIFFINV
   55 IF(IC.NE.K) STIFF(IC,K) = STIFF(IC,K)/BIGA                STIFFINV
C     REDUCE MATRIX                                             STIFFINV
      DO 65 I=1,N                                               STIFFINV
      IF(I.EQ.K ) GO TO 65                                      STIFFINV
      DO 60 J=1,N                                               STIFFINV
   60 IF(J.NE.K) STIFF(I,J) = STIFF(I,K)*STIFF(K,J)+STIFF(I,J)  STIFFINV
   65 CONTINUE                                                  STIFFINV
C     DIVIDE ROW BY PIVOT                                       STIFFINV
      DO 75 JR=1,N                                              STIFFINV
   75 IF(JR.NE.K ) STIFF(K,JR) = STIFF(K,JR)/STIFF(K,K)         STIFFINV
C     REPLACE PIVOT BY RECIPROCAL                               STIFFINV
   80 STIFF(K,K) = 1.D0/STIFF(K,K)
C     FINAL ROW AND COLUMN INTERCHANGE                          STIFFINV
      K=N                                                       STIFFINV
  100 K=K-1                                                     STIFFINV
      IF(K.LE.0) GO TO 150                                      STIFFINV
      J = ( NDEX(K) - 1) / NSZ                                  STIFFINV
      IROW =  NDEX(K) - J * NSZ                                 STIFFINV
      IF(IROW.LE.K) GO TO 120                                   STIFFINV
```

DFAL17 FORTRAN Listings (Continued)

```
      DO 130 I=1,N                                    STIFFINV
      HOLD = STIFF(I,K)                               STIFFINV
      STIFF(I,K) = -STIFF(I,IROW)                     STIFFINV
  130 STIFF(I,IROW) = HOLD                            STIFFINV
  120 JCOL =J+1                                       STIFFINV
      IF(JCOL.LE.K) GO TO 100                         STIFFINV
      DO 110 J=1,N                                    STIFFINV
      HOLD= STIFF(K,J)                                STIFFINV
      STIFF(K,J) = -STIFF(JCOL,J)                     STIFFINV
  110 STIFF(JCOL,J) = HOLD                            STIFFINV
      GO TO 100                                       STIFFINV
   25 WRITE(6,2)
    2 FORMAT('0SINGULAR STIFFNESS MATRIX')
  150 RETURN                                          STIFFINV
      END                                             STIFFINV
                        CARD COUNT        70
```

DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE MPRINT (A,M,N,MD)
C     MATRIX PRINTING ROUTINE.                                          MITE0030
C        PRINTS AN M BY N MATRIX, A , DIMENSIONED MD ROWS , ON NTAPE.   MITE0040
      DIMENSION   A(1), IT(6), C(6)                                     MITE0050
      DIMENSION FNUM(6),FFMT(8)                                         MITE0060
      DATA FFMT   /'(1H0,4X, 6(8,7HCOLUMN ,I4)/ )'/,
     1     FNUM   /'1(5X2(6X3(6X4(5X5(6X6(6X'/                          MITE0090
      EQUIVALENCE(IT(1),C(1))
      REAL*8 A
      N1=N                                                              MITE0120
      N2=6                                                              MITE0130
      N3=6                                                              MITE0140
      N4=1                                                              MITE0150
    4 IF(N3.LE.N1) GO TO 6                                              MITE0160
      N2=N1-N3+6
      N3=N1                                                             MITE0180
    6 K=0                                                               MITE0190
      DO  7  I= N4,N3                                                   MITE0200
        K=K+1                                                           MITE0210
    7   IT(K)=I                                                         MITE0220
      FFMT(2)= FNUM(N2)                                                 MITE0230
      WRITE(6     ,FFMT) (IT(I),I=1,N2)
      DO  9  I=1,M                                                      MITE0250
        K=0                                                             MITE0260
        L=MD*(N4-1)+I                                                   MITE0270
        DO  8  J=N4,N3                                                  MITE0280
          K=K+1                                                         MITE0290
          C(K)=A(L)                                                     MITE0300
    8     L=L+MD                                                        MITE0310
    9 WRITE(6     ,3)  I,(C(K),K=1,N2)
    3 FORMAT(' ',I4,2X,6F17.7))                                         MITE0100
      IF(N3.GE.N1) RETURN
      N3=N3+6
      N4=N4+6                                                           MITE0350
      GOTO  4                                                           MITE0360
      END                                                               MITE0380
                            CARD COUNT       36
```

NOT REPRODUCIBLE

DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE MULT ( NSIZE)
      COMMON/MATRS/U,HLD,A
      COMPLEX*16 U(74,74), HLD(74)
      REAL*8 A(74,74)
      DO 6 I=1,NSIZE
      DO 7 J=1,NSIZE
      HLD(J)=(0.D0,0.D0)
      DO 7 K=1,NSIZE
 7    HLD(J)= HLD(J)+ U(I,K)*DCMPLX(A(K,J),0.D0)
      DO 6 K=1,NSIZE
 6    U(I,K) = HLD(K)
 6    CONTINUE
      RETURN
      END
                         CARD COUNT        14
```

DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE PRT3
      COMMON/MATRS/U,DUM,A
      COMMON/MCOM/AM,MSB
      COMMON/CONTRL/CCON,NAERO,NSIZE,MODES,BREF,NVEL,NSIZ,ICT,VELCTY,RHO
     1,FK,NSER
      COMPLEX *16 U(74,74),HLD,DUM(74)
      REAL*8 A( 74,74),CCON,AM(150)
      INTEGER *2 MSB(150)
      REAL          VELCTY(15)
      NUM=MAX0(NAERO,1)
      DO 1  NVEL=1,NUM
      DO 217 J=1,NSIZE
      DO 216 I=1,NSIZE
  216 U(I,J) =(0.D0,0.D0)
  217 IF(NVEL.NE.1) READ(1)(A(I,J),I=1,NSIZE)
      IF(NSER.NE.0) CALL DEST
      IF(NSER.GE.0)GO TO 2
      DO 3 J=1,NSIZE
    3 WRITE(6,4)(U(I,J),I=1,NSIZE)
    4 FORMAT(//(1PD23.4))
    2 DO 225 K =1,ICT
      J=(MSB(K)-1)/NSIZE+1
      I= MSB(K)-J*NSIZE+ NSIZE
  225 U(J,I) = U(J,I) +DCMPLX(AM(K),0.D0)
      CALL MULT ( NSIZE)
      DO 14 J=2,NSIZE
      IA=J-1
      DO 14 I=1,IA
      HLD=U(I,J)
      U(I,J)=U(J,I)
   14 U(J,I)=HLD
      WRITE       (6     ,220)      RHO   ,VELCTY(NVEL)
  220 FORMAT (1H1,40X, 12H OUTPUT DATA//             '0',20X,'DENSITY=',
     1 E20.8,10X,'REDUCED VELOCITY=',E20.8)
      CALL PRT4
    1 IF(NVEL.NE.NUM) REWIND 1
      RETURN
      END                                                       DF042760
                             CARD COUNT        38
```

DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE PRT4
      COMMON /MATRS / U,  EIG,VS
      COMMON/CONTRL/CON ,IAERO,NS    ,MODES,BREF,NVEL,NSZ ,NST,VELCTY
      COMPLEX *16 U(74,74),VS(74,3), EIG(74),VECT(74),HOLD
      INTEGER *2 NC(13),IS(74)
      REAL *4 FMT(4,3),FMA(4),VELCTY(15)
      REAL *8 BIG, BT,EP(148),DREL,CON
      EQUIVALENCE(EIG(1),EP(1)),(VS(1,3),VECT(1))
      DATA SQRC PI/.1591155 /,PICON/3.7203/,FMT/' (I3,2X,2G20.7)  (I3,2X,4
     1G20.7)  (I3,2X,6G20.7) '/,IS/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,
     216,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,
     338,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,
     460,61,62,63,64,65,66,67,68,69,70,71,72,73,74/
      DREL(HOLD)=HOLD
      CALL ALLMAT(NS,NCAL)
      MMA=NCAL-1
      DO 1 J=1,MMA
      LB= J
      JA= J+1
      BIG =  DABS(DREL(EIG(J)))
      DO 6 L= JA,NS
      BT=DABS(DREL(EIG(L)))
      IF(BT.LE. BIG) GO TO 6
      BIG = BT
      LB = L
    6 CONTINUE
      IF(LB.EQ.J) GO TO 1
      HOLD = EIG(J)
      EIG(J)= EIG(LB)
      EIG(LB) =HOLD
    1 CONTINUE
      IF(NSZ.LT.0.AND.NVEL.EQ.1)WRITE(7,15)(EIG(I),I=1,NCAL)
   15 FORMAT(6D12.6)
      MODE=MIN0(MODES,NCAL)
      IF(MODE.EQ.0) GO TO 9
      WRITE (6,3) ( I, EIG(I), I=1,MODE)
    3 FORMAT ('1' / 5X,' MODE',15X,'EIGENVALUE' //(  I10,2G18.7))
      WRITE (6,4)
    4 FORMAT ('1',      50X, 'EIGENVECTORS')
      DO 16 K=1,MODE
      CALL ALLVEC (K)
      JA=1
      BIG=CDABS(VECT(1))
      DO 11 I=2,NS
      BT=CDABS(VECT(I))
      IF(BT.LE.BIG) GO TO 11
      BIG=BT
      JA=I
   11 CONTINUE
      HOLD=VECT(JA)
      DO 12 I=1,NS
   12 VECT(I)=VECT(I)/HOLD
      NCT= MOD(K,3)
      IF(NCT.EQ.0) NCT=3
      IF(NCT.EQ.3) GO TO 5
```


DFAL17 FORTRAN Listings (Continued)

```
      DO 12 I=1,NS
12    VS(I,NCT  ) = VECT(I)
      IF(K.NE.MODE)GO TO 16
5     NCB(1)=K-NCT+1
      NCB(2)= NCB(1)+1
      NCB(3)= NCB(2)+1
      DO 13 I=1,4
13    FMA(I)= FMT(I,NCT)
      WRITE(6,7) (NCB(I),I=1,NCT)
7     FORMAT ('0',21X,I2,' VECTOR',31X,I2,' VECTOR',31X,I2,' VECTOR')
      WRITE(6,FMA) (IS(I),(VS(I,J),J=1,NCT),I=1,NS)
16    CONTINUE
      WRITE (6,201)
 201  FORMAT (1H1,20X,5H MODE ,7X, 12H OMEGA (CPS), 10X, 8H DAMPING      DF040740
     1 6X, 17H VELOCITY (KNOTS) ////)
      RT=RIC(N*HREF*VELCTY(NVEL)
      DO 310 K=1,MODE
      BIG=REAL(EIG(K))
      IF(BIG.GT.0.D0)GO TO 3052
      WRITE (6,206) K
 206  FORMAT (5H MODE I4, 32H  GIVES AN IMAGINARY FREQUENCY.   )
      OMEGA=0.
      DAMP=0.
      GO TO 2
3052  OMEGA   =SUPCON/DSQRT (BIG)
      DAMP=-(K**K)/BIG
2     VELC=RT*OMEGA
310   WRITE(6,8  ) K,OMEGA,DAMP,VELC
8     FORMAT(20X,I4,3E21.8)
9     RETURN                                                             DF041020
      END                                                                DF041030
                          CARD COUNT        86
```

DFAL17 FORTRAN Listings (Continued)

BELL HELICOPTER COMPANY

```
      SUBROUTINE REDUCE(IZ)
      COMMON /RATES /REDK(16290),RED
      EQUIVALENCE (REDK(11101),FINAL(1,1))
      REAL *8 REDK,        FINAL (74,74),RED(180),PIVOT,PIVA
      INTEGER*2 IZ(74)
      NF(I)=(I*I-I)/2
      M=1
      DO 11 I=1,180
      IF(I.EQ.IZ(M)) GO TO 5
      IC=NF(I)
      PIVOT = REDK (IC+I)
      IF(I.EQ.1) GO TO 8
      IMA=I-1
      DO 6 J=1,IMA
    6 RED(J)=REDK(IC+J)
    8 DO 1 J= I,180
    1 RED(J)=REDK(NF(J)+I)
      DO 2 L=1,180
      IF(RED(L).EQ.0.D0)GO TO 2
      PIVA = RED(L)/PIVOT
      LC=NF(L)
      DO 15 K=1,L
      KT =LC+K
   15 REDK(KT) =REDK(KT)-RED(K)*PIVA
    2 CONTINUE
      GO TO 11
    5 M=M+1
   11 CONTINUE
      DO 3 J=1,74
      JC=NF(J)
      KT=IZ(J)
      KC=NF(KT)
      DO 3 I=1,J
    3 REDK(JC+I)=REDK(KC+IZ(I))
      DO 4 J=1,74
      JC=NF(J)
      DO 4 I=1,J
      KC=JC+I
      FINAL(I,J)= REDK(KC)
    4 FINAL(J,I)= REDK(KC)
      RETURN
      END
                              CARD COUNT        42
```


DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE  ROTATE
      COMMON/MATRS/REDK(16290),GAMMA,AMA,DUM(247),ALPHA(30),BETA(30),PSI
     1(30),EIB(30),EIC(30),GJ(30),AF(30),L(30),KMAT(12,12),AMAT(12,12),
     2S(12)
      EQUIVALENCE (BGPOC(1,1,1),ALPHA(1))
      REAL*8   EIB,EIC,GJ,AF,ALPHA,BETA,PSI   ,L,KMAT,AMAT,AL1,AL2,AL3,AL4
     1,SA,SB,SP,CA,CB,CP,REDK,S,SG,CG,GAMMA(30),AMA(3,3),BGPOC(3,3,60)
     2,DUM
      DO 200 I=1,16290
  200 REDK(I)=0.D0
      DO  2 J=1,12
      DO  2 I=1,12
    2 AMAT(I,J)=0.D0
      DO 5    II=1,30
      SA=DSIN(ALPHA(II))
      IF(DABS(SA).LT.1.D-15)SA=0.D0
      SB =DSIN(BETA(II))
      IF(DABS(SB).LT.1.D-15)SB=0.D0
      SP=DSIN(PSI(II))
      IF(DABS(SP).LT.1.D-15)SP=0.D0
      CA=DCOS(ALPHA(II))
      CP=DCOS(PSI(II))
      CB=DCOS(BETA(II))
      AMAT(1,1)=CA*CB*CP-SB*SA
      AMAT(1,2) =CP*SA*CB+SB*CA
      AMAT(1,3) =-CB*SP
      AMAT(2,1) =-CP*SB*CA-CB*SA
      AMAT(2,2) =CB*CA-SB*CP*SA
      AMAT(2,3) =   SB*SP
      AMAT(3,1) =   SP*CA
      AMAT(3,2) =SP*SA
      AMAT(3,3) = CP
      DO 100 K=3,9,3
      DO 100 I=1,3
      K1= I+K
      DO 100 J=1,3
      L1= J+K
      IF(DABS(AMAT(I,J)).LT.1.D-15)AMAT(I,J)=0.D0
  100 AMAT(K1,L1)=AMAT(I,J)
      DO 4 J=1,12
      DO 4 I=1,J
    4 KMAT(I,J)=0.D0
      AL1=1.D0/ L(II)
      AL4=AL1*4.D0
      AL2=  AL1* AL1 * 6.D0
      AL3=AL2*AL4*.5D0
      KMAT(1,1) =   AF(II)*AL1
      KMAT(1,7) =   -KMAT(1,1)
      KMAT(7,7) =   KMAT(1,1)
      KMAT(4,4) =   GJ(II)*AL1
      KMAT(4,10)=   -KMAT(4,4)
      KMAT(10,10)= KMAT(4,4)
      KMAT(6,6)  =       EIC(II)*AL4
      KMAT(6,12) = KMAT(6,6)*.5D0
      KMAT(12,12)= KMAT(6,6)
```

DFAL17 FORTRAN Listings (Continued)

```
      KMAT(5,5)   =      FIB(II)*AL4
      KMAT(5,11) = KMAT(5,5) *.500
      KMAT(11,11)= KMAT(5,5)
      KMAT(3,3)   = FIB(II)*AL3
      KMAT(3,9)   =-KMAT(3,3)
      KMAT(9,9)   = KMAT(3,3)
      KMAT(2,2)   = FIC(II)*AL3
      KMAT(2,8)   =-KMAT(2,2)
      KMAT(8,8)   = KMAT(2,2)
      KMAT(2,6)     =  AL2 * FIC(II)
      KMAT(2,12)    =  KMAT(2,6)
      KMAT(6,8)     = -KMAT(2,6)
      KMAT(8,12)    =  KMAT(6,8)
      KMAT(3,5)     = -AL2* EIB(II)
      KMAT(3,11)    =  KMAT(3,5)
      KMAT(5,9)     = -KMAT(3,5)
      KMAT(9,11)    =  KMAT(5,9)
      DO 1  I=1,11
      K=I+1
      DO 1  J=K,12
1     KMAT(J,I) = KMAT(I,J)
      CALL CPROD (AMAT,KMAT, 12,S)
      IF(II  .EQ.17.OR.II  .EQ.24) GO TO 3
      ISTART =II*6-12
      DO 9  J=1,12
      J1 = ISTART + J
      KC=(J1*J1-J1)/2
      DO 8  I=1,J
      I1 = ISTART + I
      IF(I1.LE.0.OR.J1.LE.0) GO TO 8
      REDK(KC+I1)=REDK(KC+I1)+KMAT(I,J)
8     CONTINUE
      GO TO 9
3     DO 11 J=1,6
      J1 = J + 6
      J2 = J + 48
      J3 = J + 96
      IF(II  .EQ.24) J3=J+138
      JC=(J2*J2-J2)/2
      JD=(J3*J3-J3)/2
      DO 5 I=1,6
      I2=I+48
5     REDK(JD+I2)=KMAT(I,J1)
      DO 11 I=1,J
      I1 = I + 6
      I2 = I + 48
      I3 = I + 96
      IF(II  .EQ.24) I3=I+138
      JT=JC+I2
      REDK(JT)= REDK(JT)+ KMAT(I,J)
      KT=JC+I3
11    REDK(KT)=REDK(KT)+KMAT(I1,J1)
9     CONTINUE
      DO 6  I=1,30
6     GAMMA(I)=-PSI(I)
```

DFAL17 FORTRAN Listings (Continued)

```
      AMA (3,2)  = 0.D0
      AMA (2,3)  = 0.D0
      AMA (2,2)  = 1.D0
      AMA (2,1)  = 0.D0
      AMA (1,2)  = 0.D0
      DO 7 II=1,30
      SG=DSIN(GAMMA(II))
      IF(DABS(SG).LT.1.D-15)SG=0.D0
      CG = DCOS(GAMMA(II))
      AMA (1,1)  = CG
      AMA (1,3)  = SG
      AMA (3,1)  = -SG
      AMA (3,3)  = CG
      DO 7 J=1,3
      DO 7 I=1,3
      RGRCC(J,I,2*II) = AMA (I,J)
    7 RGRCC(J,I,2*II-1) = AMA (I,J)
      RETURN
      END
                              CARD COUNT      129
```

DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE STGN(NZS)
      COMMON/CONTRL/DUMA(7),NSIZ,DUMB(17),FK
      COMMON/MATES/ FINAL(74,74),DUM(5624),STIF(74,74),ALPHA
     1(30),BETA(30),PSI(30),EIB(30),EIC(30),GJ(30),AF(30),L(30),PARM
      REAL*8   EIB,EIC,GJ,AC,ALPHA,BETA,PSI,A,L,DUMF,    DUMF,STIF,FINAL
     1,PARM(22,5),PI,SPT,PAPA(22,5)
      REAL LTPVLS,MVCLST,LCA,XDL(5,2),XDEIB(5,2),XDEIC(5,2),XDGJ(5,2)
      INTEGER*2 I7(74)
      EQUIVALENCE (PARM(1,1),PAPA(1,1))
      DATA I7/3,4,9,10,15,16,21,22,27,28,44,45,46,49,67,68,69,70,73,74,
     175,76,79,80,81,82,85,86,87,88,91,92,93,94,109,110,111,112,115,116,
     2117,118,121,122,123,124,127,128,129,130,133,134,135,136,151,152,
     3153,154,157,158,159,160,163,164,165,166,169,170,171,172,175,176,
     4177,178 /,PI/3.14159265358979320D0/
      SPT=PI/3.D0
      A=1.D0/FK
      READ (5,11) WGSWEP,BLDSWP,VRPVLS,LTPVLS,FOLDSF,XMASTF
   11 FORMAT(6E12.5)
      READ (5,11) FXCLST,MVCLST,LCA,PVLOFF,H,EHUB
      WRITE(6,15) WGSWEP,BLDSWP,LCA,PVLOFF,H,EHUB,VRPVLS,LTPVLS,FOLDSF,
     1FXCLST,MVCLST ,XMASTF
   15 FORMAT('1FLEXIBILITY MATRIX DATA'
     1//' WING SWEEP',F43.5,' (DEGREES)'/' BLADE SWEEP' F42.5,' (DEGREES
     2)'/' DIST-FA TO CONVERSION AXIS',F25.3,' (INCHES)'/' PYLON OFFSET'
     3,F41.5,' (INCHES)'/' MAST LENGTH',F42.5,' (INCHES)'/' HUB OFFSET',
     4 F43.5,' (INCHES)'/' PYLON-F/A STIFFNESS',F33.1,' (IN-LBS/RADIAN)
     5'/' PYLON-LAT STIFFNESS',F33.1,' (IN-LBS/RADIAN)'/' BLADE FOLD STI
     6FFNESS',F29.1,' (IN-LBS/RADIAN)'/' FIXED CONTROL STIFFNESS',F26.1,
     7' (IN-LBS/RADIAN)'/' MOVABLE CONTROL STIFFNESS',F24.1,' (IN-LBS/RA
     8DIAN)'/' MAST STIFFNESS',F35.1,' (IN-LBS/RADIAN)')
      WGSWEP=WGSWEP*SPT/6.D1
      BLDSWP=BLDSWP*SPT/6.D1
      DO 101J=1,2
      DO 101 I=1,5
  101 READ (5,11) XDL(I,J),XDEIB(I,J),XDEIC(I,J),XDGJ(I,J)
      DUM=XDEIB(2,1)*1.E6
      DO 120  LZ=1,30
      BETA(LZ)=0.D0
      ALPHA(LZ)=0.D0
      PSI  (LZ)=0.D0
      EIB(LZ)=DUM
      EIC(LZ)=DUM
      L(LZ)=1.D0
  120 AF(LZ) =DUM
      DO 106 I=11,16
      BETA(I-10)=WGSWEP
  106 BETA(I)   =BLDSWP
      DO 107 I=17,30
      BETA(I)   =-BLDSWP
      ALPHA(I) =PI
      PSI(I)=SPT
  107 IF(I.GE.24) PSI(I)=-SPT
      ALPHA(7)  = PI*.5D0
      ALPHA(9)  =-ALPHA(7)
      BETA (24)=0.D0
```


DFAL17 FORTRAN Listings (Continued)

```
      BETA (17) =0.D0
      DO 109 I=1,5
      L(I) = XDL(I,1)
      EIB(I) = XDEIB(I,1)
      EIC(I) =  XDEIC(I,1)
      GJ(I)  =  XDGJ(I,1)
      DO 109 K=11,25,7
      M=I+K
      L(M) = XDL(I,2)
      EIB(M) =  XDEIB(I,2)
      EIC(M) =  XDEIC(I,2)
      L(K-1)=EHUB
      GJ(K-1)=EHUB*FKCLST
      EIC(K)=FULDSF
      GJ(K) =MVCLST
  109 GJ(M)  =  XDGJ(I,2)
      L(6)=.5*L(5)
      L(7) =LCA
      L(8) = PYLOFF
      L(9) = H
      EIC(6) = EIC(5)
      EIC(8)=PYLOFF**2*LTPYLS*.5
      EIB(6) = EIB(5)
      GJ(6) = GJ(5)
      GJ(7) = DUM
      GJ(8) = PYLOFF*VAPYLS
      GJ(9) = KMASTF*H
      CALL ROTATE
      CALL TPROD
      CALL REDUCE(IZ)
      READ (5,11) ((PARM(I,J),I=1,12),J=1,5)
      DO 3 I=11,12
      IP1=I+1
      DO 3 J=1,NZS
      DUME=STIF(I,J )
      STIF(I,J)=STIF(IP1,J)
    3 STIF(IP1,J)=DUME
      DO 5 I=1,NZS
      DO 16 J=1,NZS
   16 FINAL(I,J)=0.D0
      DO 4 J=11,12
      DUME=STIF(I,J )
      STIF(I,J)=STIF(I,J+1)
    4 STIF(I,J+1)=DUME
    5 FINAL(I,I)=1.D0
      CALL INVERS(NZS)
      MVCLST=A   /STIF(12,12)
      LTPYLS=A   /STIF(11,11)
      WRITE(6,21)LTPYLS,MVCLST,(I,XDL(I,1),XDEIB(I,1),XDEIC(I,1),XDGJ(I,
     11),(PARM(I,J),J=1,5),I=1,5)
   21 FORMAT (' BEAMWISE STIFFNESS(CONVERSION AXIS)',F14.1,' (LBS/IN)'/
     1' CHORDWISE STIFFNESS(CONVERSION AXIS)',F13.1,' (LBS/IN)'// 'CWING
     2 PARAMETERS'//' SEG', 9X,'LENGTH', 9X,'EIB',11X,'EIC',12X,'GJ',12X
     4,'K',13X,'V',13X,'M',13X,'S',13X,'T' //(I3,4X, 9F14.5))
      WRITE (6,1) ((PARM(I,J),J=1,5),I=6,7),(I,XDL(I,2),XDEIB(I,2),XDEIC
```


DFAL17 FORTRAN Listings (Continued)

```
      1I,2),XDGJ(I,2), (PARA(I,J),J=1,5),I=1,5)
 1    FORMAT( / /'PYLON PARAMETERS'//16X,'F',13X,'V',13X,'M',13X,'S',
     1 13X,'I'//' PITCH',2X, 5E14.6/' YAW',4X,5E14.6//'BLADE PARAMETERS
     2'//' SEG',9X,'LENGTH',9X,'FIB',11X,'FIC',12X,'GJ',12X,'K',13X,'V',
     313X,'M',13X,'S',13X,'I'/ /(I3,4X,9F14.5))
      DO 20 J=1,5
      DO 20 I=8,12
      PARM(I+5,J)=PARM(I,J)
 20   PARM(I+10,J)=PARM(I,J)
      I=-1
      M=2
      DO 17 J=1,22
      IF(J.EQ.8) M=4
      I=I+M
      IP1=I+1
      FINAL(I,IP1)=1.D0
      FINAL(IP1,I)=-PARM(J,1)
 17   FINAL(IP1,IP1)= PARM(J,2)
      CALL DPROD(FINAL,STIF,NZS,EIB)
      IF(NSIZ.GT.0) GO TO 7
      NCT=0
      NZT=NZS-6
      DO 18 J=1,NZS
      DO 6 K=1,NZT,6
      KPF=K+5
      NCT=NCT+1
 6    WRITE(7,2) (STIF(J,I),I=K,KPF),NCT
 2    FORMAT(6D12.6,I4)
      NCT=NCT+1
 18   WRITE(7,8) STIF(J,NZS-1),STIF(J,NZS),NCT
 8    FORMAT(2D12.6,48X,I4)
 7    RETURN
      END
                              CARD COUNT       143
```

DFAL17 FORTRAN Listings (Continued)

```
      SUBROUTINE TPROD
      COMMON/MATRS/B,XTD,XTE,DUM(269),XTC
      REAL *8 B(16290),XTP,DUM
     1,XTC(3,3,60),XTD(3,3   ),XTE(3,3)
      NF(L)=(L*L-7*L)/2+3*L+J
      DO 1 I=1,60
      DO 1 J=1,60
      ISC = (9*J*J-21*J)/2+3*I+3
      IF(I.EQ.J) GOTO 3
      DO 8 K=1,3
      ISP=ISC+K
      DO 8 L=1,3
    8 XTE(K,L) =B(ISP+NF(L))
      GO TO 10
    3 DO 11 K=1,3
      ISP=ISC+K
      DO 11 L=K,3
      XTE(K,L) =B(ISP+NF(L))
   11 XTE(L,K) =XTE(K,L)
   10 DO 2 K=1,3
      DO 2 M=1,3
      XTD(K,M   ) =0.D0
      DO 2 L=1,3
    2 XTD(K,M   ) = XTD(K,M   ) +XTC(K,L,I)*XTE(L,M)
      DO 1  L=1,3
      ISP=ISC+L
      DO 1  M=1,3
      IF(L.EQ.J.AND.L.GT.M) GO TO 1
      XTP=0.D0
      DO 15 K=1,3
   15 XTP=XTP+XTD(L,K)*XTC(M,K,J)
      B(ISP+NF(M))=XTP
    1 CONTINUE
      RETURN
      END
                              CARD COUNT          35
```

DFAL17 FORTRAN Listings (Concluded)

```
      COMMON/MATRX/EV,G,Q,TTL,GV,B,DUM(21960),AMK
      COMMON/MATRA/AM,C
      COMMON/MATRC/        TOT,NPTS,NR,YHT,VG,NI,NS
      COMMON/MATRD/  DT,OM  ,NW
      COMMON/MATRE/EY ,NSB
      COMMON /MATRF/ C,FD,AMS,DUMA(40),PC,VBR,NP,NV,ICK
      REAL*8 TOT,P1A,PI,FK , PC(10,2),CA  ,DT,FV(74),FY(74),TL,U(10)
     1,AMK(74,74),CS,DC,CD       ,G(10),        YC(1003,6),C(2,2,22)
     *,E1,GJ,STA,SLC,SLE,SLAS,XHS,     AT,CCA,CDE,CLC,COD,SLAT
      LOGICAL*1 STOP
      INTEGER*2 NSD(4),  NSG(  22),NSB(4),NSC0(10),ICK(6)
      REAL*4  BUF( 51,3), TTL(20),AM(150),DV(22),B(22),XD(1003,3)
     1, EI(10),GK(10),SLC(10),SLD(10)
      COMPLEX*16 C(74),AWE(6,6,10),FM(10,6)
      EQUIVALENCE (EV(1),YC(1,1)),(YC(2749,6),XD(1,1))
      DATA NZS /74/,PI/3.14159265358979327,       NSG/1,1,3,5,7,11,
     1 13,17,17,21,25,29,33,37,41,45,45,53,57,61,65,69/ ,NSD/1,2,5,6/
      CALL ERRSET(208,0,-1,0,0,0)
      PIA=PI/180.
      CALL PLOTS(BUF,512)
3     READ(5,12,END=11) TTL
12    FORMAT (20A4)
      CALL SYMBOL(0.,   .5,.4,TTL,90.,80 )
      CALL PLOT (1.,1.,-3)
      READ(5,4         ) NI,NT,NP,NSC,(NSC0(I),I=1,NSC)
4     FORMAT(16I4)
      NG= IABS(NI)
      WRITE (6,27) TTL, NI,NT,NP,NSC
27    FORMAT ('1',40X,'GUST RESPONSE ANALYSIS FOR FOLDING PROPROTOR'//'0
     1',20A4//'CONTROL DATA'//'0GUST NUMBER',I22/' OPTION NUMBER',I20/'
     2 NO. OF FLUTTER FREQUENCIES',I7/' NO. OF SECTIONS',I18)
      READ(5,1) RHO,V,CLA,VG,BR,S,DT,OM,TMAX,FK,WGSWEP,BLDSWP
1     FORMAT (6E12.5)
      READ (5,15) (EI (I),GK(I),SLC(I),SLD(I),I=1,NSC)
15    FORMAT (4E12.5)
      WRITE (6,17) RHO,V,CLA,VG, BR,S, DT, OM,TMAX,FK,WGSWEP,BLDSWP,
     1(NSC0(I), EI(I), GK(I),SLC(I), SLD(I), I=1,NSC)
17    FORMAT(' DENSITY,SLUG/FT**3', E15.5/' AIRCRAFT VELOCITY,KT',E13.5/
     1' LIFT CURVE SLOPE',E17.5/' GUST VELOCITY',E20.5 /' REF. SEMICHORD
     2,FT',E16.5 /' SPAN,FT', E26.5 /' TIME INCREMENT,SEC', E15.5 /' GUS
     3T FREQUENCY,CPS',E15.5 /' MAX. RUN TIME,SEC',E16.5 / ' FLEXIBILITY
     4 FACTOR', E15.5 /' WING SWEEP,DEG',E19.5 /' BLADE SWEEP,DEG',E18.5
     5//'0SECTION PARAMETERS' //' SECTION',9X,'EI0,LB-IN**2',13X,'GJ,LB-
     6IN**2',16X,'L1,IN',19X,'L2,IN'//(I5,4E24.5))
      NPTS = MINO ( IFIX (TMAX/SNGL(DT ))+1,1001)
      DO 30 I=1,NZS
30    FV(I)=0.D0
      TDT = 2. /DT
      V=V*1.688
      VER=V/BR
      GST= RHO*V*CLA*VG
      READ(5,1) DV
      READ(5,1) B
      WRITE(6,5)(I,DV(I),B(I),I=1,22)
5     FORMAT ( //20X,'STRIP WIDTHS(GUST)', 5X,'SEMICHORDS(GUST)' //(I3,
```


DFAL18 FORTRAN Listings

```
     1E31.5,E22.5))
      J=-1
      IV=2
      DO 18 I=1,22
      IF(I.EQ.8) IV=4
      J=J+IV
 18   FV(J    ) = CSI*DV(I)*B(I)
      READ(5,1)(G(I),U(I),I=1,NP)
      WRITE(6,9) (I,G(I),U(I),I=1,NP)
 9    FORMAT(// '1INPUT FREQUENCY-DAMPING VALUES' //' MODE',20X,'DAMPING
     1',14X, 'FREQUENCY' //(I3,18X,E13.5,9X,E13.5))
      CALL EFAC(RHO*BR**2*S)
      CALL MASS(ICT,NZS)
      STOP=.FALSE.
      NCT=0
      NZM=NZS-6
      DO 7 I=1,NZS
      DO 10 K=1,NZM,6
      KPF=K+5
      NCT=NCT+1
      READ(5,21)(AW(I,J),J=K,KPF),IKT
 21   FORMAT(6D12.5,I4)
      IF(IKT.EQ.NCT) GO TO 10
      WRITE(6,13) IKT
 13   FORMAT('0FLEX CARD',I4,' OUT OF ORDER')
      NCT=IKT
      STOP=.TRUE.
 10   CONTINUE
      NCT=NCT+1
      READ(5,20) AW(I,NZS-1),AW(I,NZS),IKT
 20   FORMAT (2D12.5,48X,I4)
      IF(IKT.EQ.NCT) GO TO 7
      WRITE(6,13) IKT
      NCT=IKT
      STOP=.TRUE.
 7    CONTINUE
      READ(5,1 )(U(I),I=1,NZS)
      DO 25 I=1,NP
      PC(I,2)=2.D0*PI*U(I)
      PC(I,1)=G(I)*PC(I,2)
      DA=1.D0/PC(I,2)**2
 25   W(I)=CMPLX(DA,G(I)*DA)
      WRITE(6,6) (I,W(I),I=1,74)
 6    FORMAT(//'1FLUTTER SYSTEM ROOTS'//(I15,E14.5,E13.5,I7,E14.5,E13.5,
     1I7,E14.5,E13.5))
      WRITE(6,26 )
 26   FORMAT ('1FLEXIBILITY MATRIX,IN/LB' //)
      DO 16 I=1,NZS
      WRITE(6,2 )  (AW(I,J),J=1,NZS)
 2    FORMAT('0',D11.4,10D12.4/(11D12.4))
      FV(I)=0.D0
      DO 16 J=1,NZS
      AW(I,J) =AW(I,J)/12.*FK
 16   FV(I)=FV(I)+AW(I,J)*FV(J)
      IF(STOP) GO TO 3
```

DFAL18 FORTRAN Listings (Continued)

```
      WRITE(6,14) TTL
   14 FORMAT('1',20A4/'OSYSTEM RESPONSES')
      DO 22 ISC=1,NSC
      NS=NSCE(ISC)
      ICK(1)=NSG(NS)
      DO 29 J=1,4
      NSB(J)=NSD(J)
      IF(NS.LT.6) NSB(J)=J
   29 ICK(J+1)=ICK(1)+J
      ICK(6)=ICK(5)+1
      NV=4
      IF(NT.EQ.0.OR.NS.EQ.6.OR.NS.EQ.7) NV=2
      CALL RESCCM(NZS,6   ,ICT)
      NSK=MAX0(NS-1,1)
      IF(NS.GT.5.AND.NS.LT.9) NSK=NS
      DO  8 L=2,NV,2
      LA=L-1
      DO 43 J=1,NPTS
      DA=C(1,1,NSK)*YC(J,LA)+C(1,2,NSK)*YC(J,L)
      YC(J,L)=C(2,1,NSK)*YC(J,LA)+C(2,2,NSK)*YC(J,L)
   43 YC(J,LA)=DA
    8 NSK=NSK+1
      NN=2
      IF(NV.EQ.2)GO TO 59
      EIB= EI(ISC)/1.44D2
      GJ = GK(ISC)/1.44D2
      SLA = SLC(ISC)/1.2D1
      SLB = SLD(ISC)/1.2D1
      SLE = SLB -SLA
      SLAS=SLA**2
      XHS = SLB**2
      TL =2.D0/SLE
      CCD = 3.D0/SLE
      CCA= TL*EIB
      CCC = GJ/SLE
      CCB = CCD*CCA
      DA=SLAS*XHS*SLE
      DB=XHS*(3.D0*SLAS-2.D0*SLA*SLB)/DA
      DC=SLAS*(2.D0*SLA*SLB-3.DC*XHS)/DA
      SLAT=SLAS**2/DA
      XT=XHS**2/DA
      GT= BLDSWP*PIA
      IF(NS.LT.6) GT=WGSWEP*PIA
      GC=  COS(GT)
      GS=  SIN(GT)
      DO 54 I=1,NPTS
      YC(I,5)= YC(I,1)*DB-YC(I,3)*SLAT
      IF(NS.NE.1) GO TO 57
      YC(I,6)=YC(I,5)
      YC(I,5)=0.
      YC(I,3)=YC(I,1)
      YC(I,4)=YC(I,2)
      YC(I,1)=0.
      YC(I,2)=0.
      GO TO 58
```

DFAL18 FORTRAN Listings (Continued)

```
57    YC(I,6)= YC(I,1)*XT     +YC(I,3)*DC
58    YC(I,2)= (YC(I,2)+GS*YC(I,5))/GC
      YC(I,4)= (YC(I,4)+GS*YC(I,6))/GC
      XC(I,1)= COE*(TL  *(YC(I,3)-YC(I,1))+YC(I,5)+YC(I,6))
      XC(I,2)= COE*(YC(I,4)-YC(I,2))
54    XC(I,3)= COA*(COD*(YC(I,3)-YC(I,1))+2.*YC(I,6)+YC(I,5))
      DO 19 J=1,3
      NR=J+1
19    CALL HIST(XD(1,J))
      IF(NT.EQ.1) GO TO 22
      NN=6
59    DO 60 J=1,NN
      NN=J+6
      DO 55 I=1,NPTS
55    XC(I,1)=YC(I,J)
60    CALL HIST(XD(1,1))
22    CALL PLOT(12.,-VHT-1.,-3)
      GO TO 3
11    CALL PLTIME
      CALL PLOT(0.,0.,999)
      STOP
      END
                         CARD COUNT       187
```

DFAL18 FORTRAN Listings (Continued)

```
      SUBROUTINE DFAL(CON)
      COMMON/PAT15/ A,CHI,PARTA,AM,BM,EJOX,EYOX
      REAL    COSLMA(4),BRA(4),SA(4),CAPA(4),CBAA(4)
     1,AM(2,2  ),BM(2,4),B(7,4),EYOX(2),EJOX(2),DELTAV(7,4),D(7,4)
      REAL*8 CHI(2,4),PARTA(2,4)
      COMPLEX*16 A(4,7,4)
      INTEGER*2  ISZB(4)
      DATA N/2/,M/4/,PI/3.141593/,NPRO/4/,ISZE/7,5,5,5/
      WRITE(6,4)
    4 FORMAT(//'OAERODYNAMIC INPUT DATA' /// '  SURFACE',2X,'COSINE LAMB
     1DA',10X,'BR',15X,'S',14X,'CAP S',12X,'C BAR' // )
      DO 2  JKL=1,NPRO
      READ(5,3) COSLMA(JKL),BRA(JKL),SA(JKL),CAPA(JKL),CBAA(JKL)
    3 FORMAT (6E12.8)
      ISZ=ISZB(JKL)
      WRITE(6,1) JKL, COSLMA(JKL),BRA(JKL),SA(JKL),CAPA(JKL),CBAA(JKL)
    1 FORMAT (I6,5E17.5)
      READ(5,3)(DELTAV(I,JKL),I=1,ISZ)
      READ(5,3) (B(I,JKL),I=1,ISZ)
    2 READ(5,3) (D(I,JKL),I=1,ISZ)
      WRITE (6,9)
    9 FORMAT(//9X,'STRIP',9X,'1',16X,'2',16X,'3',16X,'4',16X,'5',16X,'6'
     1,16X,'7'/)
      DO 10JKL=1,NPRO
      IF(ISZB(JKL)-6)6,10,11
    6 WRITE(6,5) (DELTAV(I,JKL),I=1,5),JKL,(B(I,JKL),I=1,5),
     1 (D(I,JKL),I=1,5)
    5 FORMAT('OSURFACE  DV',5E17.5/I5,6X,'B',5E17.5/11X,'D',5E17.5)
      GO TO 10
   11 WRITE(6,7) (DELTAV(I,JKL),I=1,7),JKL,(B(I,JKL),I=1,7),
     1 (D(I,JKL),I=1,7)
    7 FORMAT('OSURFACE  DV',7E17.5/I5,6X,'B',7E17.5/11X,'D',7E17.5)
   10 CONTINUE
      RETURN
      ENTRY DF67(EKR,J)
      IST=-1
      IV=2
      DO 60 JKL=1,NPRO
      ISZ=ISZB(JKL)
      SCACON=PI/SA(JKL)*COSLMA(JKL)/BRA(JKL)**2*CON
      DO 60 I=1,ISZ
      IF(IST.EQ.13) IV=4
      IST=IST+IV
      SCALER    =SCACON*B(I,JKL)**2*DELTAV(I,JKL)
      BID=B(I,JKL)/D(I,JKL)
      AM(1,1  ) =1.
      AM(1,2  ) =-BID
      AM(2,1  ) =0.
      AM(2,2  ) = BID
      EIK=EKR    *BRA(JKL)/B(I,JKL)
      ERNOW =1. /EIK
      CALL BESY (ERNOW,      EYOX   , IEY)
      DO 8 K=1,2
      CALL BESJ (ERNOW,K-1, EJOX(K),1.E-5,IEJ)
    8 IF(IEJ+IEY.NE.0) WRITE(6,12) JKL,J,I,K, ERNOW
```

NOT REPRODUCIBLE

DFAL18 FORTRAN Listings (Continued)

```
 12   FORMAT (' ERROR IN BESSEL FUNCTION,SURFACE',I3,'  VELOCITY',I3,'
     1 STRIP',I3,'  ORDER', I3,'  X VALUE', F12.5)
      DEN=(EJUX(2)*EVOX(1))**2+(EVOX(2)-EJOX(1))**2
      F=(EJUX(2)*(EJUX(2)*EVOX(1))*EVOX(2)*(EVOX(2)-EJOX(1)))/DEN
      G= -(EVOX(2)*EVOX(1)+EJUX(2)*EJUX(1))/DEN
      EK2 =EIK*EIK
      TSA=2.*G*EIK
      TSB=2.*F*EIK
      BP(1,1)=1.+TSA
      BP(1,2)=-TSB
      BP(1,3) = .5*TSA-2.*F*EK2
      BP(1,4) = -EIK-TSB- 2.*G*EK2
      BP(2,1) =.5
      BP(2,2) =0.
      BP(2,3) =.375
      BP(2,4) =-EIK
      DO 62 K=1,N
      DO 62 L=1,N2
      PART A(K,L) =0.D0
      DO 62 J=1,N
 62   PART A(K,L) = PART A(K,L)+AM(K,J  )* BM(J,L)
      DO 61 L=1,N2
      M=(L+1)/2
      LM=MOD(L,2)
      DO 61 K=1,N
      CHI (K,L  )=0.D0
      DO 63 J=1,N
 63   CHI (K,L  ) =CHI(K,L  ) +PART A  (K,2*JJ-LM)*AM(M,JJ)
 61   CHI(K,L)=CHI(K,L)*SCALEN
      ISA=IST-1
      DO 60 K=1,N
      ISK=ISA+K
      M=0
      DO 60 L=1,N2,2
      M=M+1
 60   A(ISA+M,ISK)=DCMPLX(CHI(K,L),CHI(K,L+1))
      RETURN
      END                                                  H9070203
                         CARD COUNT        93
```

DEAL18 FORTRAN Listings (Continued)

BELL HELICOPTER COMPANY

```
      SUBROUTINE MASS(ICT,NZS)
      COMMON/MATRS/P,X,Y,M,SX,IX,C,B,DD
      COMMON/MATRA/AM,CM,MSB
      REAL*8          X(22),Y(22),M(22),SX(22),IX(22),C(2,2),B(2,2),DD(2)
     1,CM(2,2,22),PARM(22,5),A(120)
      REAL*4 AM(150)
      INTEGER *2 MSB(150)
      EQUIVALENCE (PARM(1,1),A(1))
      READ (5,2)((PARM(I,J),I=1,12),J=1,5)
    2 FORMAT (6E12.0)
      DO 1 J=1,5
      DO 1 I=18,22
      PARM(I,J)= PARM(I-10,J)
    1 PARM(I-5,J)= PARM(I,J)
      WRITE(6,3) (I,X(I),Y(I),M(I),SX(I),IX(I),I=1,22)
    3 FORMAT( '1MASS INPUT DATA'//' SEGMENT',15X,'X',16X,'Y', 16X,'M',15
     1X,'SX',15X,'IX'//(I5,7X, 5E17.5))
      ICT=0
      NV=-2
      DO 7 I=1,22
      C(2,2) =  1.D0/(X(I)+Y(I))
      C(1,2) =X(I)*C(2,2)
      C(1,1)= Y(I)*C(2,2)
      C(2,1) =  -C(2,2)
      DO 35 K=1,2
      CM(2,K,I)=C(2,K)*12.D0
   35 CM(1,K,I)=C(1,K)
      B(1,1) =  M(I)
      B(1,2) =  SX(I)
      B(2,1) =  SX(I)
      B(2,2) =  IX(I)
      IF(I.LT.9) GO TO 11
      DO 13 K=1,2
      NV=NV+1
      IC= I+I+NV
      ICT=ICT+1
      AM(ICT)=M(I)/32.174
   13 MSB(ICT)= IC*NZS+IC-NZS
   11 IC= I+I+NV
      CALL DPROD(C,B,2,DD)
      DO 7 K=1,2
      ICK=K+IC
      DO 7 J=1,2
      IF(B(J,K).EQ.0.D0) GO TO 7
      ICT=ICT+1
      IF(ICT.GT.150) STOP
      MSB(ICT)=NZS*ICK+J+IC-NZS
      AM(ICT)= B(J,K)/32.174
    7 CONTINUE
      RETURN
      END
                        CARD COUNT        51
```

NOT REPRODUCIBLE

DFAL18 FORTRAN Listings (Continued)

```
      SUBROUTINE BESJ(X,N,BJ,D,IER)                               DF040380
      BJ=0.                                                       DF040390
      IER=1
      IF(N)200,20,20
   20 IER=2
      IF(X)200,200,31
   31 IF(X-15.) 32,32,34                                          DF040460
   32 NTEST= 20.+10.*X-X**2/3.                                    DF040470
      GO TO 36                                                    DF040480
   34 NTEST= 90.+X/2.                                             DF040490
   36 IF(N-NTEST) 40,38,38                                        DF040500
   38 IER=4                                                       DF040510
      GO TO 200
   40 IER=0                                                       DF040530
      BPREV=0.                                                    DF040550
      IF(X-5.) [illegible],60,60                                  DF040590
   50 MA=X+6.                                                     DF040600
      GO TO 70                                                    DF040610
   60 MA=1.4*X+ [illegible]                                       DF040620
   70 MZERO=MAX [illegible] +IFIX(X)/4+2)
      MMAX=NT[illegible]                                          DF040680
      DO 190 [illegible],MAX,3
      FM1=1 [illegible]                                           DF040730
      FM=0                                                        DF040740
      ALPH[illegible]                                             DF040750
      IF(M [illegible]0,110,120
  110 JT=-1                                                       DF040770
      GO TO [illegible]                                           DF040780
  120 JT=1                                                        DF040790
  130 M2=M-2                                                      DF040800
      DO 160 K=1,M2                                               DF040810
      MK=M-K                                                      DF040820
      BMK=2.*FLOAT(MK)*FM1/X-FM                                   DF040830
      FM=FM1                                                      DF040840
      FM1=BMK                                                     DF040850
      IF(MK.EQ.N+1)BJ=BMK
      JT=-JT
  160 ALPHA=ALPHA+BMK*FLOAT(JT+1)
      BMK=2.*FM1/X-FM                                             DF040910
      IF(N.EQ.0)BJ=BMK
      ALPHA=ALPHA+BMK
      BJ=BJ/ALPHA                                                 DF040950
      IF(ABS(BJ-BPREV)-ABS(D*BJ)) 200,200,190                     DF040960
  190 BPREV=BJ                                                    DF040970
      IER=3                                                       DF040980
  200 RETURN                                                      DF040990
      END                                                         DF041000
                        CARD COUNT        47
```

DFAL18 FORTRAN Listings (Continued)

```
      SUBROUTINE BESY(X,  BY,IER)                                        
      REAL BY(2)
      IER=1
      IF(X)170,170,20
 20   IER=0
      IF(X-4.0)40,40,30
 30   T1=4.0/X                                                            BESY 550
      T2=T1*T1                                                            BESY 560
      P0=((((-.0000037043*T2+.00001735651)*T2-.00004876131)*T2           BESY 570
     1 +.00017343)*T2-.001753062)*T2+.3989423                             BESY 580
      Q0=((((.00000032312*T2-.0000142078)*T2+.0000342468)*T2             BESY 590
     1 -.0000869791)*T2+.0004564324)*T2-.01246694                         BESY 600
      P1=((((.0000042414*T2-.0000200920)*T2+.0000580759)*T2              BESY 610
     1 -.0002232031)*T2+.002921826)*T2+.3989423                           BESY 620
      Q1=((((-.0000036594*T2+.00001622)*T2-.0000398708)*T2               BESY 630
     1 +.0001064741)*T2-.0006390400)*T2+.03740084                         BESY 640
      A=SQRT(T1)
      B=A*T1                                                              BESY 660
      C=X-.7853982                                                        BESY 670
      SC= SIN(C)
      CC= COS(C)
      Y0=A*P0*SC    +B*Q0*CC
      Y1=-A*P1*CC    +B*Q1*SC
      GO TO 90                                                            BESY 700
 40   XX=X/2.                                                             BESY 740
      XT=XX*XX
      T=ALOG(XX)+.5772157                                                 BESY 760
      SUM=0.                                                              BESY 770
      TERM=XT*(1.-T)
      Y0=T*TERM
      TERA = XX*(T-.5)
      Y1=TERA
      DO 80 L=2,16
      FL1=L-1
      SUM=SUM+1./FL1
      FL=L                                                                BESY 920
      TS=T-SUM
      TERM= TERM*(XX/FL)**2*(1./(FL*TS)-1.)
      Y0=Y0+TERM
      TERA=TERA*XT    /(FL1*FL   )*(.5/FL-TS)/(TS+.5/FL1)
 80   Y1=Y1+TERA
      PI2=.6366198                                                        BESY 970
      Y0=PI2*Y0                                                           BESY 980
      Y1=PI2*(Y1-1./X)
 90   BY(1)=Y0
      BY(2)=Y1
 170  RETURN                                                              BESY1290
      END                                                                 BESY1340
                              CARD COUNT        48
```


DFAL18 FORTRAN Listings (Continued)

```
      SUBROUTINE OPROD(A,B,N,S)
      REAL *8 A(N,N), B(N,N), S(N),SAVE
      NM1=N-1
      DO 100 I=1,2
      IF(I.EQ.1) GO TO 1
      DO 10 II=1,NM1
      III = II + 1
      DO 10 IK=III,N
      SAVE = B(II,IK)
      B(II,IK) = B(IK,II)
 10   B(IK,II) = SAVE
 1    DO 100 J=1,N
      DO 50 K=1,N
      S(K) = 0.D0
      DO 50 L=1,N
 50   S(K)= S(K) + B(J,L)*A(L,K)
      DO 100 KJ=1,N
 100  B(J,KJ) = S(KJ)
      RETURN
      END
                              CARD COUNT         20
```

DFAL18 FORTRAN Listings (Continued)

```
      SUBROUTINE FN (NX)
      COMMON/MATRS/DUM(12036),X
      COMMON/MATRD/  DT,OMI,NG
      COMMON/MATRF/DUMA(2056),VAR
      REAL *8      CM,SCT,CDT,X(1003),DT
      X(1)=0.D0
      NXM=NX
      IF(NG.EQ.1) GO TO 3
      IF(NG.NE.2) GO TO 6
      NXM=.5D0/(OMI*DT)+1.
    3 CM=OMI*6.2831853
      CMT=CM*DT
      SDT = SIN(CMT)
      CDT = COS(CMT)
      SCT= 0.D0
      CCT= 1.D0
      DO 1 I=2,NXM
      X(I)= SCT*CDT+CCT*SDT
      CCT = CCT*CDT-SCT*SDT
    1 SCT = X(I)
      IF(NG.EQ.1) GO TO 5
      DO 7 I=2,NXM
    7 X(I)=X(I)**2
      NXT=NXM+1
      GO TO 2
    8 IF(NG.NE.5)GO TO 9
      DO 10 I=2,NX
      T=-DFLOAT(I-1)*DT*VAR
   10 X(I)=1.-.165*EXP(.0455*T)-.335*EXP(.3*T)
      GO TO 5
    9 DO 6 I=2,NX
    6 X(I)=1.D0
      IF(NG.EQ.3) GO TO 5
      NXT=CMI/DT+1.
    2 DO 4 I=NXT,NX
    4 X(I)=0.D0
    5 CALL TMF
      RETURN
      END                          CARD COUNT        39
```

DFAL18 FORTRAN Listings (Continued)

```
      SUBROUTINE INVDET (N,P)
      COMMON/MATRS/DUM,STIFF
      REAL*8 LOAD
      COMPLEX*16 STIFF(74,74),     BIGC,P,DUM(74,74)
      CDABS(BIGC)=BIGC*DCONJG(BIGC)
      P=(1.D0,0.D0)
      DO 1 K=1,N
      BIGC =-STIFF(K,K)
      IF(CDABS(BIGC).EQ.0.D0) STOP
      P=P*STIFF(K,K)
      IF(K.EQ.N)GO TO 1
      L=K+1
      DO 55 IC=L,N
   55           STIFF(IC,K) = STIFF(IC,K)/BIGC
      DO 60 I=L,N
      DO 60 J=L,N
   60           STIFF(I,J) = STIFF(I,K)*STIFF(K,J)+STIFF(I,J)
    1 RETURN
      END
                              CARD COUNT      19                        STIFFINV
```

DFAL18 FORTRAN Listings (Continued)

```
      SUBROUTINE MULT ( NSIZE)
      COMMON/MATRS/U,HLD,A
      COMPLEX*16 U(74,74),HLD(74)
      REAL*8 A(74,74)
      DO 8 J=1,NSIZE
      DO 7 I=1,NSIZE
      HLD(I)=(0.D0,0.D0)
      DO 7 K=1,NSIZE
    7 HLD(I)= HLD(I)+ U(J,K)*DCMPLX(A(K,I ),0.D0)
      DO 8 K=1,NSIZE
    8 U(J,K)=HLD(K)
      RETURN
      END
                          CARD COUNT        13
```

DFAL18 FORTRAN Listings (Continued)

```
      SUBROUTINE REDUCE(N,M,L)
      COMMON /MATRS/ A,F
      COMMON/MATRF/DM(74),FB,AK,FR,DUMS(43),IZ
      INTEGER*2 IZ(6),IR(74)
      COMPLEX*16  A(74,74), AI,       FB(10,6),AK(6,6,10),AP,FI,F(74),DM
     1,PR(10)
      REAL*8 CDAB,CEB,CEM
      CDAB(AP)=AP*DCONJG(AP)
      DO 1 J=1,N
    1 IR(J)=J
      DO 10 J=1,M
      IF(IZ(J).NE.L)PR(L)=PR(L)/(DM(IZ(J))-DM(L))
   10 IR(IZ(J))=0
    9 I=0
      CEB=0.D0
      DO 5 J=1,N
      IF(IR(J).EQ.0)GO TO 5
      CEM=CDAB(A(J,J))
      IF(CEM.LE.CEB)GO TO 5
      CEB=CEM
      I=IR(J)
    5 CONTINUE
      IF(I.EQ.0) GO TO 8
      IR(I)=0
      PR(L)=PR(L)*A(I,I)
      IF(I.NE.L)PR(L)=PR(L)/(DM(I)-DM(L))
      AI=(1.D0,0.D0)/A(I,I)
      FI=F(I)*AI
      DO 2 J=1,N
      F(J)=     F(J)-FI*A(J,I)
    2 A(I,J)=A(I,J)*AI
      F(I)=(0.D0,0.D0)
      DO 3 J=1,N
      AP=-A(J,I)
      IF(J.EQ.I .OR.CDAB(AP).EQ.0.D0) GO TO 3
      DO 4 K=1,N
    4 IF(K.NE.I)A(J,K)=A(J,K)+AP*A(I,K)
    3 CONTINUE
      DO 7 J=1,N
      A(J,I)=(0.D0,0.D0)
    7 A(I,J)=(0.D0,0.D0)
      GO TO 9
    8 DO 6 J=1,M
      IJ=IZ(J)
      FB(L,J)=F(IJ)
      DO 6 I=1,M
      II=IZ(I)
    6 AK(I,J,L)=A(II,IJ)
      RETURN
      END
                              CARD COUNT        50
```


DFAL18 FORTRAN Listings (Continued)

```
      SUBROUTINE RESCOM(NZS,NRS,ICT)
      COMMON /MATRS/A,Z
      COMMON /MATRA/AM,C,NSB
      COMMON /MATRC/ CA,CB,NPTS
      COMMON /MATRE/ FY,NSB
      COMMON /MATRF/ Q,FB,AMB,PR,    PC,VBR,NP,NV
      EQUIVALENCE (Z(1),AM(1,1)),(YC(25(9,6),XF(1,1,1)),(A(1,1),YC(1,1))
      COMPLEX*16 A(74,74),Z(74),PH(10),C(74),FB(10,6),AMB(6,6,10),SMP,PD
     1,A=(6,6)
      REAL*8 FY(74),PC(10,2),DREL,YC(100,6),C(2,2,2),XB(10,6,4)
      REAL*4 AM(150)
      INTEGER*2          NSE(4),NSB(150)
      DREL(PC)=PD
      DO 8 NAR=1,NP
      DO 2 J=1,NZS
      DO 2 I=1,NZS
    2 A(I,J)=(0.D0,0.D0)
      CALL DEGT(SNGL(VBR/PC(NAR,2)),NAR)
      DO 40 K=1,ICT
      J=(NSE(K)-1)/NZS+1
      I=NSE(K)-J*NZS+NZS
   40 A(J,I)=A(J,I)+DCMPLX(DBLE(AM(K)),0.D0)
      CALL MULT(NZS)
      DO 20 I=2,NZS
      IA=I-1
      DO 20 J=1,IA
      SMP=A(I,J)
      A(I,J)=A(J,I)
   20 A(J,I)=SMP
      PH(NAR)=(1.D0,0.D0)
      DO 21 I=1,NZS
      A(I,I)=A(I,I)-C(NAR)
   21 Z(I)= DCMPLX(FY(I),0.D0)
    8 CALL REDUCE(NZS,NRS,NAR)
      CALL FN(NPTS)
      DO 48 IZ = 1,NV
      NSK= NSB(IZ)
      DO 31 NAR=1,NP
      DO 23 I=1,NRS
      DO 10 J=1,NRS
   10 AB(I,J)=AMB(I,J,NAR)
   23 AB(I,NSK)=FB(NAR,I)
      CALL INVDET(    NRS,PD)
      PD=FL*PH(NAR)
      XB(NAR,4,IZ)=0.D0
      XB(NAR,5,IZ)=0.D0
      XB(NAR,6,IZ)=0.D0
      XB(NAR,2,IZ)=PC(NAR,2)**2
      XB(NAR,1,IZ)= DREL(PD)*XB(NAR,2,IZ)
   31 XB(NAR,3,IZ)= PC(NAR,1)
   48 CALL RESP(YC(1,IZ),XB(1,1,IZ),NP)
      RETURN
      END
                              CARD COUNT        53
```


DFAL18 FORTRAN Listings (Continued)

```
      SUBROUTINE RESP(Y   ,X,IP)
      COMMON/MATRS/DUM(6018),Z
      COMMON/MATMC/TOT,NT
      REAL*8 DUP,TOT,CA,CT    ,Y(1003),Z(1003),X(10,6),TOTN
      DO 13 I=1,NT
      Y(I)=0.D0
      DO 13 IN=1,IP
      TOTN=TOT*X(IN,3)
      CT=TOT*X(IN,5 )*X(IN,6 )
      CA=TOT*X(IN,4)*X(IN,5)
      X(IN,4 )=(X(IN,1)*Z(I)+CA*TOTN+CT)/(TOTN*TOT+X(IN,2))
      X(IN,5 ) = TOT*X(IN,4 )-CA
      X(IN, 6 ) = TOT*X(IN,5 )-CT
   13 Y(I)=Y(I)+X(IN,4)
      RETURN
      END
                              CARD COUNT        16
```

DFAL18 FORTRAN Listings (Continued)

BELL HELICOPTER COMPANY

```
      SUBROUTINE TMP
      COMMON/PATHC/TUT      ,NPTS,NR,YHT,VG,NI,NS
      COMMON/MATRD/DT
      COMMON/PATHS/COM(12000),X,U,W
      REAL*8 A(100),DT,TLT,TTL(2,10)
      DIMENSION U(100),V(100),W(100)
      INTEGER*2 NCR(7),NUMB(22),NUT(6),N(10)
      DATA TTL /'GUST INPUT,FPS  FORCE,LB        TORQUE,FT-LB    ANGLE,
     1FT-LB   Z LEFT,FT        TWIST LEFT,RAD  Z RIGHT,FT       TWIST -RI
     2HT,RAD SLOPE LEFT,RAD  SLOPE RIGHT,RAD '/,N/1,3,2,12,9,1,6,1,15
     3,14,15/,NUT/'COORDINATE  '/,NUMB/'01020304050607080910111213141516
     417181920212 2'/
      YHT=0.
      DO 4 I=1,6
    4 NCR(I)=NUT(I)
      NCR(7)=NUMB(NS)
      IF(NI.LT.0) WRITE(6,13) NCR
   13 FORMAT(//'0',7A2)
      NPA= NPTS+1
      NPB=NPA+1
      DO 1 I=1,NPTS
      W(I)=X(I)*VG
    1 U(I)=FLOAT(I-1)*DT
      CALL SYMBOL(3.5,-.8,.2,NCR,0.,14)
      CALL SCALE(U,10.0,NPTS,1,20.0)
      CALL SCALE(W,2.0,NPTS,1,20.0)
      CALL AXIS(0.0,0.0,'TIME(SEC)',-9,10.0,0.0,U(NPA),U(NPB),20.)
      CALL AXIS(0.,0.,TTL(1,1),14,2.,90.,W(NPA),W(NPB),20.)
      CALL LINE(U,W,NPTS,1,0,0)
      RETURN
      ENTRY HIST(V)
      IF(NI.LT.0.OR.NR.LT.5) WRITE(6,2)(TTL(I,NR),I=1,2),(V(I),I=1,NPTS)
    2 FORMAT(//'0',2A8//(10E13.5))
      YHT=YHT+2.75
      NRR=N(NR)
      CALL PLOT( 0.0,2.75   ,-3)
      CALL SCALE(V,2.0, NPTS,1,20.0)
      CALL AXIS(0.,0.,TTL(1,NR),NRR  ,2.,90.,V(NPA),V(NPB),20.)
      CALL LINE(U,V,NPTS,1,0,0)
      RETURN
      END
                              CARD COUNT        41
```

NOT REPRODUCIBLE

DFAL18 FORTRAN Listings (Concluded)

BELL HELICOPTER COMPANY

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Bell Helicopter Company Fort Worth, Texas 76101	2a. REPORT SECURITY CLASSIFICATION Unclassified
	2b. GROUP

3. REPORT TITLE

A STUDY OF FOLDING PROPROTOR VTOL AIRCRAFT DYNAMICS -- VOLUME II

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report

5. AUTHOR(S) (First name, middle initial, last name)
H. E. Losey
P. Y. Hsieh

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1971	244	

8a. CONTRACT OR GRANT NO. F33615-69-C-1339	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. 1370	
c. Task No. 137005	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	AFFDL-TR-71-7, VOLUME II

10. DISTRIBUTION STATEMENT
Distribution limited to U. S. Government agencies only, test and evaluation; statement applied 12 August 1971. Other requests for this document must be referred to AF Flight Dynamics Laboratory (FY) Wright-Patterson AFB, Ohio 45433.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Air Force Flight Dynamics Laboratory Air Force Systems Command Wright-Patterson Air Force Base, Ohio

13. ABSTRACT

This report describes the results of a study of the dynamic stability and response during the feathering and folding of the blades of a folding-proprotor VTOL. This study involved the development of analytical methods of predicting the dynamic characteristics during feathering and folding, correlation of the theoretical methods with experimental data, a dynamic analysis of a representative folding-proprotor VTOL design, and a parametric study to identify design factors which can be used to control dynamic characteristics during feathering and folding. Correlation of the theory with measured dynamic stability and response characteristics during feathering and folding is good, and indicates the analytical methods can be used with confidence. The results of the dynamic analysis indicate satisfactory stability and response characteristics can be achieved in a 66,000-pound gross-weight class, folding-proprotor VTOL. The results of the parametric study are summarized in terms of design guidelines. This volume is a guide to the computer programs and contains FORTRAN listings.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Folding Proprotor Dynamic Stability Dynamic Response Feathering/Unfeathering Propeller Whirl Flutter Reduced Frequency						

☆U.S.Government Printing Office: 1971 — 759-078/147

UNCLASSIFIED
Security Classification